APCO P25 Security Revisited

Matt Robert

About Myself

- Amateur Security Researcher
- Licensed Amateur Radio Operator
- Systems Infrastructure and Platform manager as my day job
- Aerobatic pilot when I have the spare time (and money…)
- And no – I don’t always look like David Boone or Chopper Read imitator – its for “Movember” ☺

Presentation Overview

- What this presentation will cover:-
  - A brief intro into to Software radio and how the OP25 software works
  - A description of some of the practical flaws the P25 protocol
  - and some of the fixes too!

Presentation Overview

- What this presentation *IS NOT*…
  - A guide to hacking your local Public Safety P25 communications system ☺
  - Going to distribute tools or software that help achieve this aim..

OP25 Project

- Website is www.sedition.org.au
- Mailing list is on yahoogroups – OP25-dev list.
  - About 200 members
  - About 8 developers
    - Max Parke – KA1RBI - DSP, signal processing, demodulators, etc
    - Stephen Glass – VK4SMG - Project founder and leader
    - Michael Ossmann – Wrote wireshark plugin
    - Myself – VK2TVK – Project coordinator, publicist and researcher.
    - Balint Seeber VK2FUNK – contributed DSE-OFB decryption support

OP25 Project – Proudly supported by NICTA!

NICTA

Australian Government

Department of Broadband, Communications and the Digital Economy

Australian Research Council

NICTA Funding and Supporting Members and Partners

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

UNSW THE UNIVERSITY OF NEW SOUTH WALES

NSW GOVERNMENT Industry & Investment

Victoria The Place To Be

THE UNIVERSITY OF MELBOURNE

THE UNIVERSITY OF SYDNEY

Queensland Government

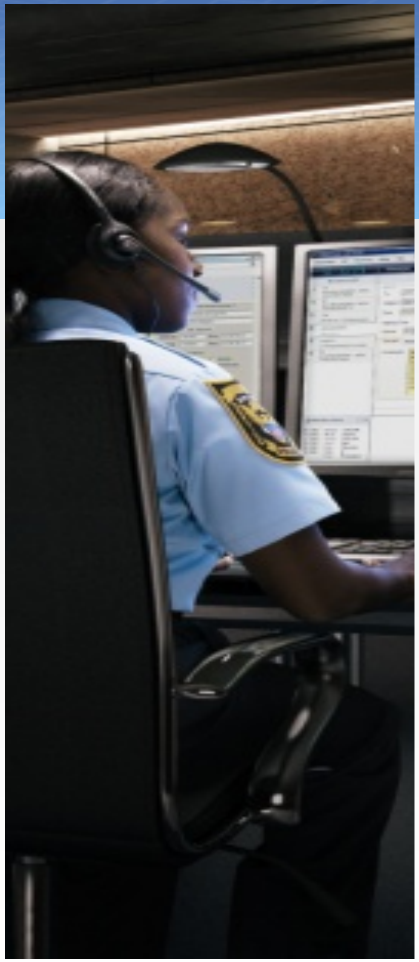
Griffith UNIVERSITY

QUT Queensland University of Technology

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

P25 Users

- Various Australia Federal law enforcement agencies
- NSW, QLD, SA State Police forces
- Airport Fire brigades around Oz
- Victorian MMR – All emergency services in metro area are on a P25 trunked network
- NSW GRN – All NSW Emergency services now using APCO25 trunking as well
- SA GRN - All metro SA Emergency services also migrated to P25



Photographs Source: Nivas Iver, Radio Authentication Customer Presentation, Motorola, 2006

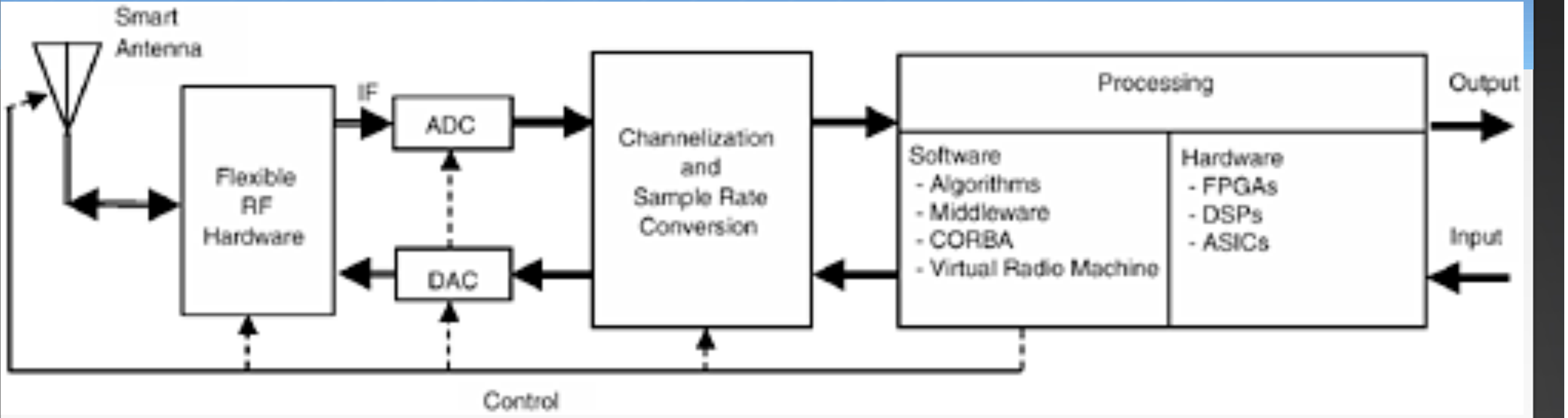
Why should *we* care?

- Ensure that public safety agencies and officers can continue to do their jobs safely
- Make sure that law enforcement and public safety agencies can have private, secure communications
- Protect public information from being disseminated (personal details – drivers license number, DOB, street address, etc)

And it’s not just us that’s concerned

- A group of researchers in the US have started looking into P25 also.
- It’s a team headed up by two guys, Sandy Clark and famous cryptographer Matt Blaze..
- Matt Blaze said a few months ago:-
- "It's going to be someone somewhere creating the Project 25 jamming kit and it'll be something that you download from the Net," Blaze said. "We're not there right now, but we're pretty close.”

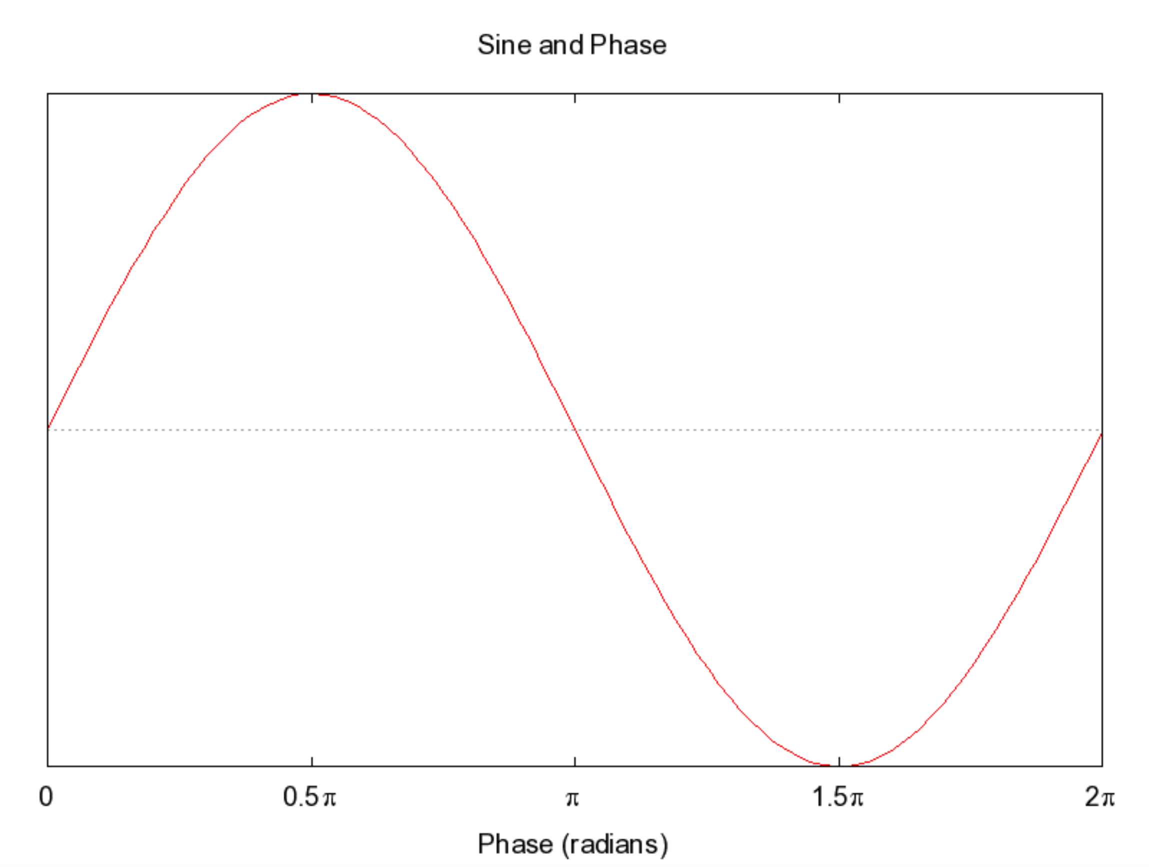
Software-Defined Radio



Source: The Scientist' and engineers guide to digital signal processing, 2nd ed, Steven W. Smith

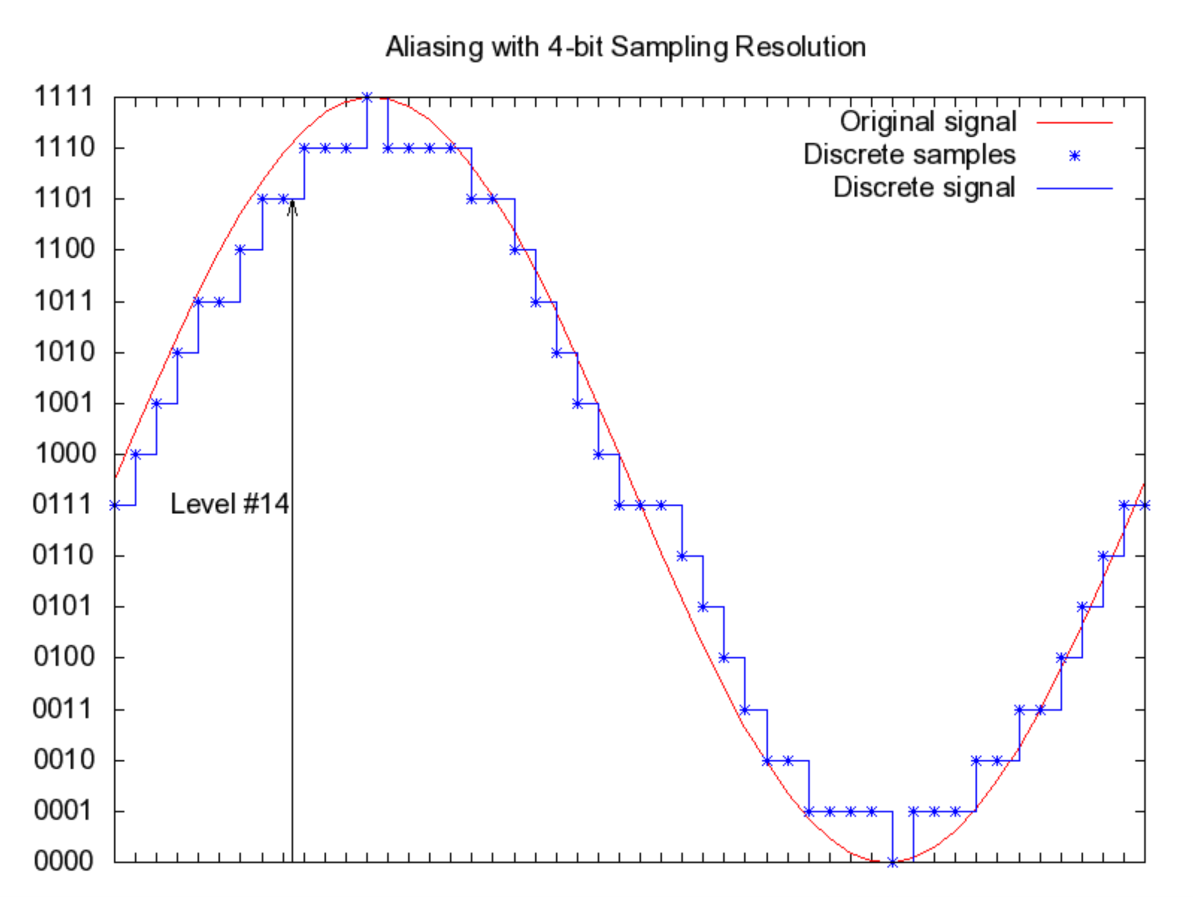
- Receiver:
  - RF down-converter
  - ADC
  - Signal Processing
- Transmitter
  - Signal Generation
  - DAC
  - RF up-converter

Sampling – Here’s our Source Signal



Copyright Neil Carter - http://psy.swan.ac.uk/staff/carter/unix/gnuplot-guide.htm

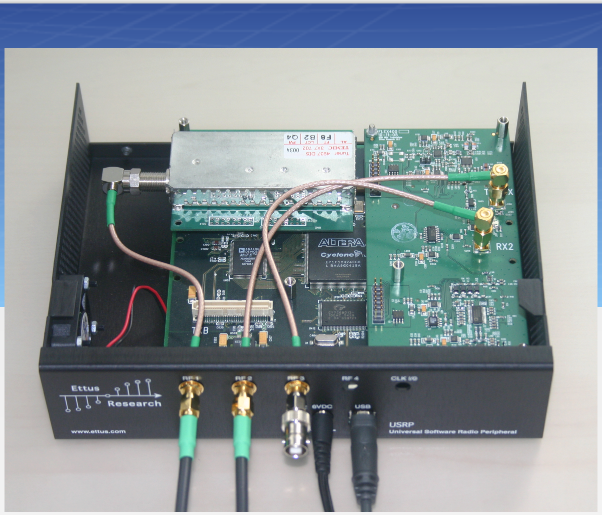
Discrete Time Series Sampling

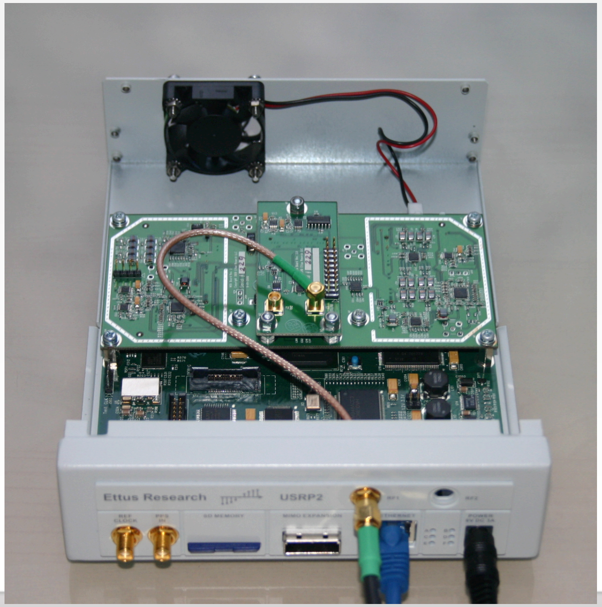


Copyright Neil Carter - http://psy.swan.ac.uk/staff/carter/unix/gnuplot-guide.htm

Sampling via USRP

- USRP1
  - 8 MHz bandwidth
  - 6,400 P25 channels at once!
  - Limited by USB 2.0 speed
  - About $1200 USD with WBX wideband RF transceiver daughterboard (50MHz-2.2GHz)
- USRP N210
  - 25MHz bandwidth
  - 20,000 P25 channels at once!
  - Gigabit Ethernet instead of USB 2.0
  - Limiting factor is usually in host processing power/throughput
  - About $2000 USD with with WBX wideband RF transceiver daughterboard (50MHz-2.2GHz)





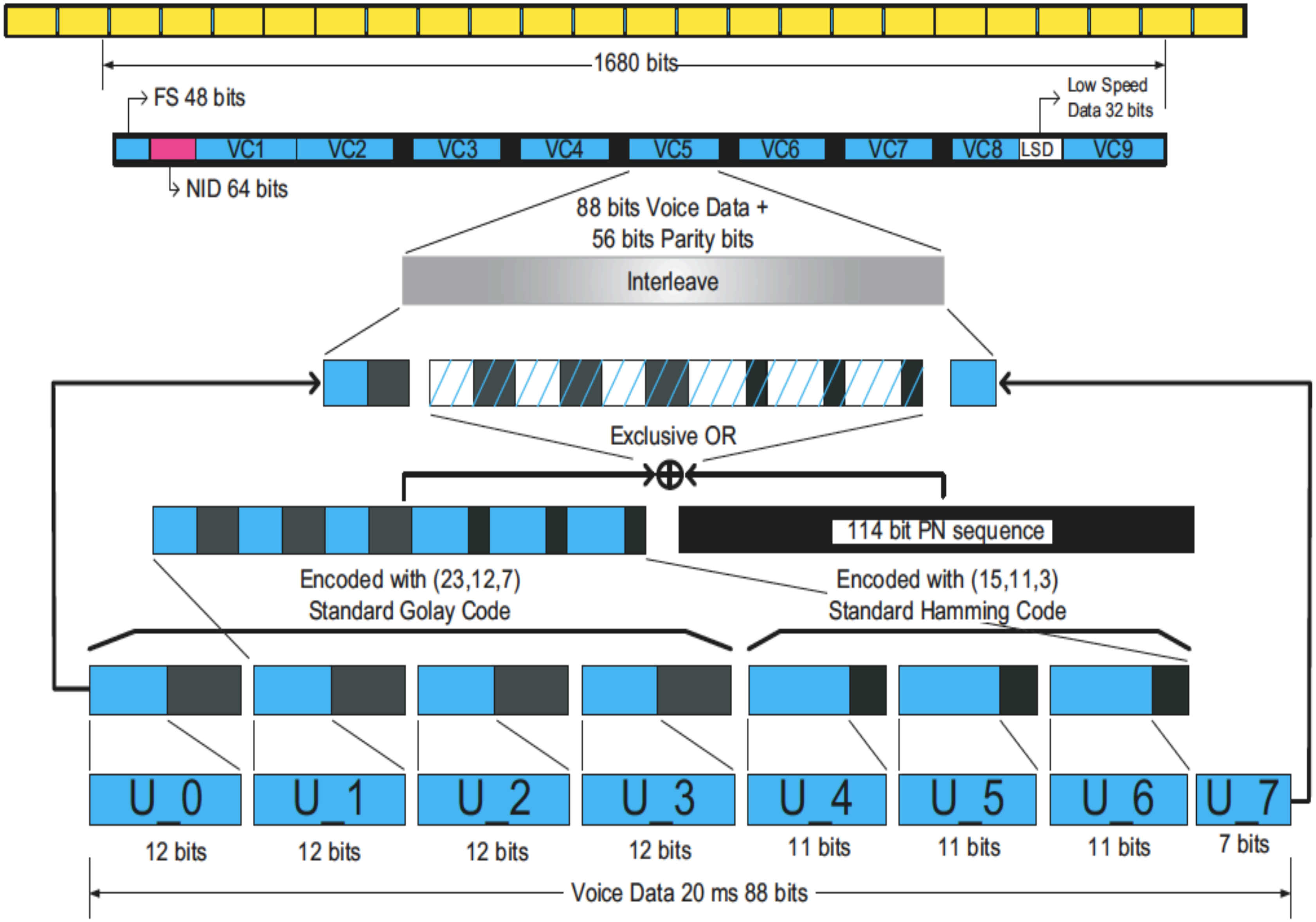
Cryptanalysis of P25

- The basis of the crypto flaws in P25 Is the good old fashioned “Known Plaintext Attack”

- Basically we are looking for a part of the message that has been encrypted and we know the encrypted contents..
  - During WW2, this was known as a “crib” at the UK’s Bletchley Park which was the UK’s main decryption and cryptanalysis establishment.
  - Back then, the allies’ cryptanalysts would look for known words in known locations, such as the german word for weather
  - Due to the regimented style of military communications there were lots of cribs in the intercepted encrypted transmissions.

OK, but how does this relate to P25?

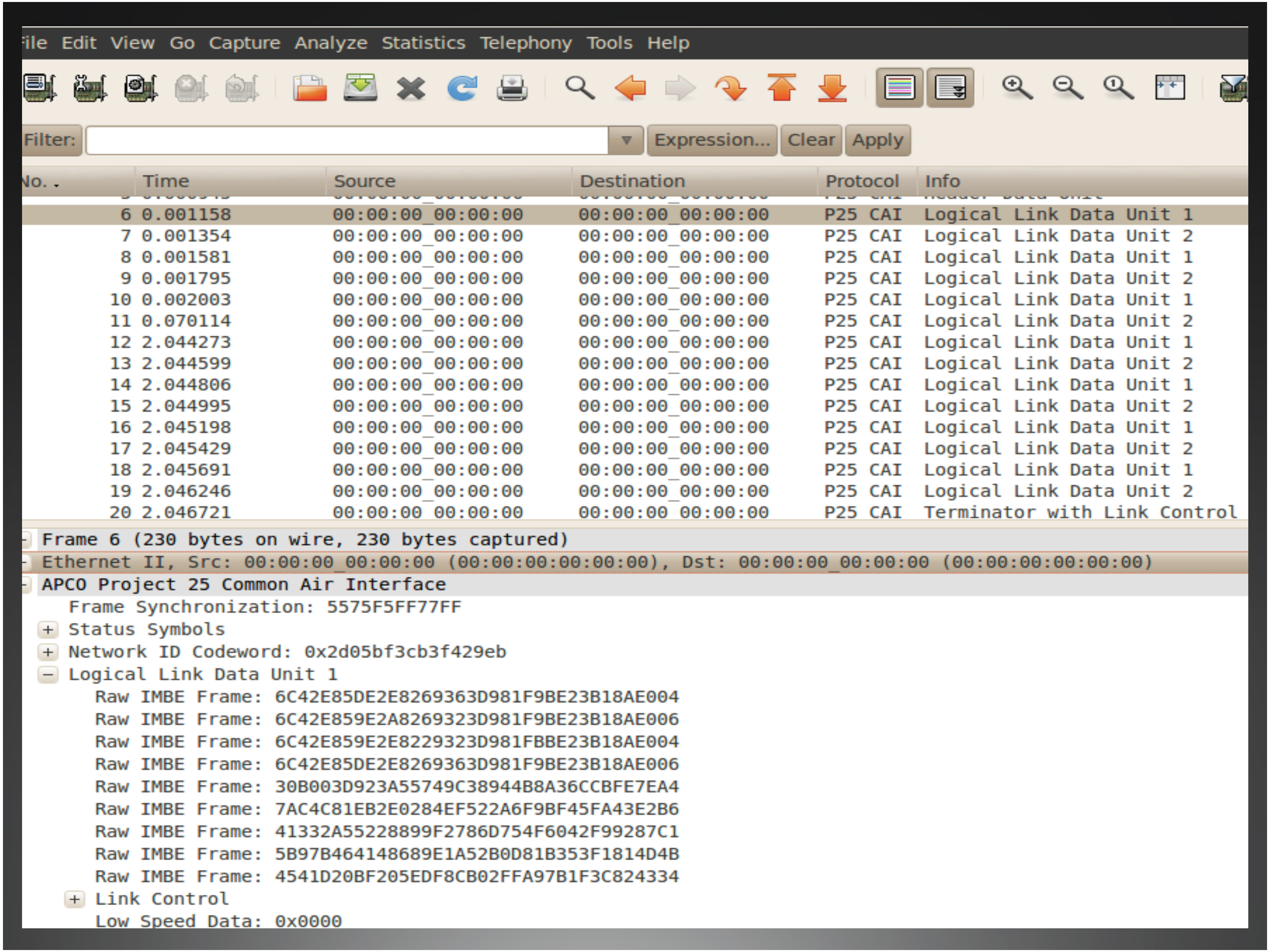
- First we need to look at how P25 sends voice traffic:-
  - 4800 Baud signal, with 4FSK (C4FM) style modulation which gives an aggregate data rate of 9600 bits per second
  - When P25 is encoding voice traffic, the IMBE Voice CODEC slices the voice traffic up into 20ms slices, which are then grouped into two types of frames called LDU1 and LDU2
  - Each frame contains 9 ‘slices’ which are actually known as IMBE codewords. Each vector contains pitch and intensity information, voiced and unvoiced noise information amongst other things..
  - These vectors are protected by varying amounts of Forward Error Correction (FEC) depending on how important the data is that is being protected..



Courtesy of Daniels Corporation P25 Training guide

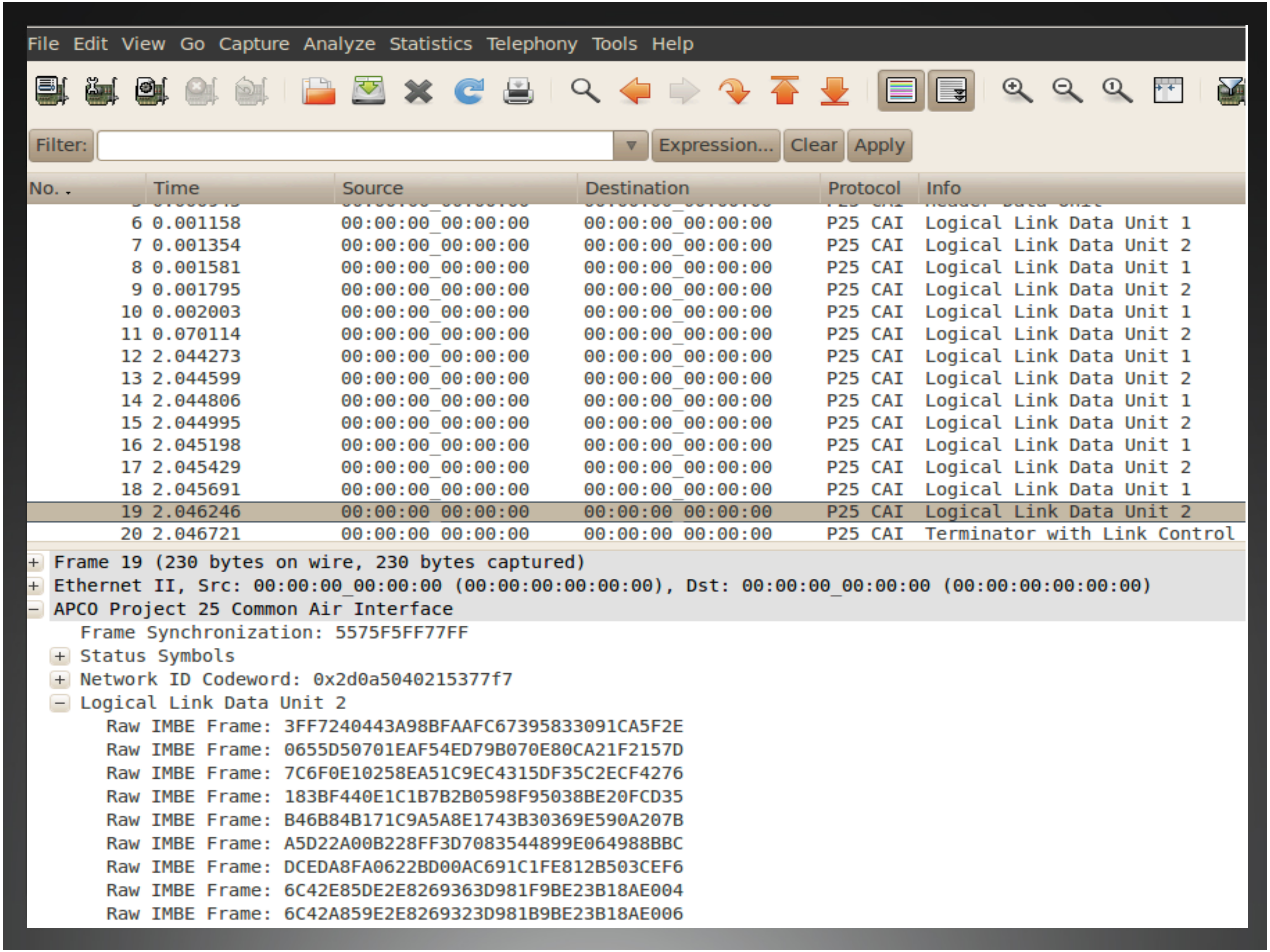
…..aaaand….?

- Well, lets take special note that voice traffic is split up into 180ms frames..
- When someone makes a transmission, they obviously have to end it at some point when they've finished speaking right?
- Well, when that happens, you have a 1 in 9 chance of being right at a frame boundary… which tells us that 8 times out of 9 you will be somewhere else.. And when you dekey your radio at that point, the radio inserts standard defined silence frames!
- And that is our version of that Known Plaintext Attack!



```
No. .        Time      Source               Destination          Protocol  Info
           6 0.001158  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
           7 0.001354  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
           8 0.001581  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
           9 0.001795  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          10 0.002003  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
          11 0.070114  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          12 2.044273  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
          13 2.044599  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          14 2.044806  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
          15 2.044995  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          16 2.045198  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
          17 2.045429  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          18 2.045691  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
          19 2.046246  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
          20 2.046721  00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Terminator with Link Control
```

```
- Frame 6 (230 bytes on wire, 230 bytes captured)
- Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- APCO Project 25 Common Air Interface
     Frame Synchronization: 5575F5FF77FF
   + Status Symbols
   + Network ID Codeword: 0x2d05bf3cb3f429eb
   − Logical Link Data Unit 1
        Raw IMBE Frame: 6C42E85DE2E8269363D981F9BE23B18AE004
        Raw IMBE Frame: 6C42E859E2A8269323D981F9BE23B18AE006
        Raw IMBE Frame: 6C42E859E2E8229323D981FBBE23B18AE004
        Raw IMBE Frame: 6C42E85DE2E8269363D981F9BE23B18AE006
        Raw IMBE Frame: 30B003D923A55749C38944B8A36CCBFE7EA4
        Raw IMBE Frame: 7AC4C81EB2E0284EF522A6F9BF45FA43E2B6
        Raw IMBE Frame: 41332A55228899F2786D754F6042F99287C1
        Raw IMBE Frame: 5B97B464148689E1A52B0D81B353F1814D4B
        Raw IMBE Frame: 4541D20BF205EDF8CB02FFA97B1F3C824334
      + Link Control
        Low Speed Data: 0x0000
```

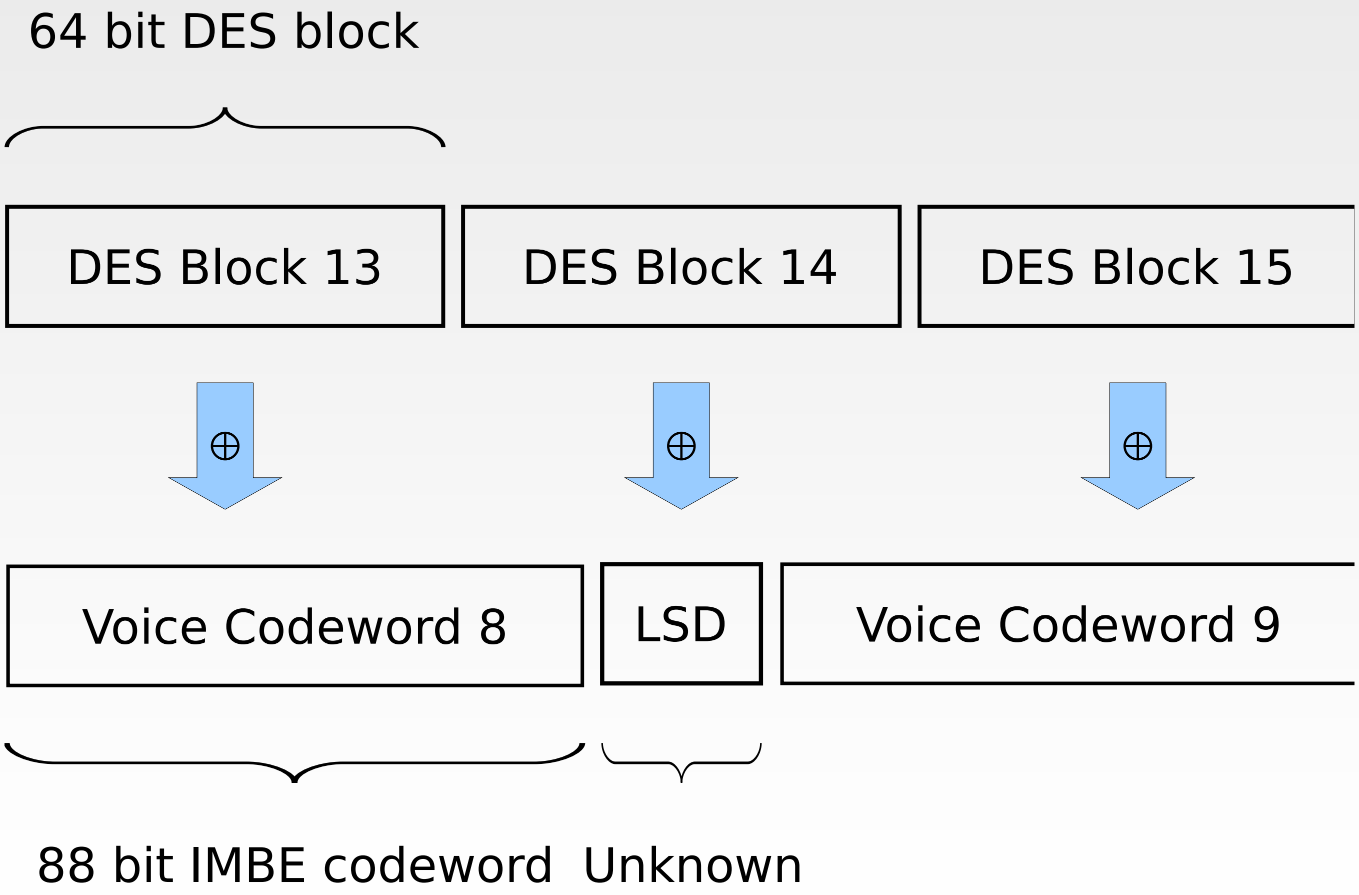


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply

```
No. .         Time           Source               Destination          Protocol  Info
        6 0.001158      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
        7 0.001354      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
        8 0.001581      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
        9 0.001795      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       10 0.002003      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
       11 0.070114      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       12 2.044273      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
       13 2.044599      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       14 2.044806      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
       15 2.044995      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       16 2.045198      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
       17 2.045429      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       18 2.045691      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 1
       19 2.046246      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Logical Link Data Unit 2
       20 2.046721      00:00:00_00:00:00    00:00:00_00:00:00    P25 CAI   Terminator with Link Control
```

```
+ Frame 19 (230 bytes on wire, 230 bytes captured)
+ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- APCO Project 25 Common Air Interface
    Frame Synchronization: 5575F5FF77FF
  + Status Symbols
  + Network ID Codeword: 0x2d0a5040215377f7
  - Logical Link Data Unit 2
      Raw IMBE Frame: 3FF7240443A98BFAAFC67395833091CA5F2E
      Raw IMBE Frame: 0655D50701EAF54ED79B070E80CA21F2157D
      Raw IMBE Frame: 7C6F0E10258EA51C9EC4315DF35C2ECF4276
      Raw IMBE Frame: 183BF440E1C1B7B2B0598F95038BE20FCD35
      Raw IMBE Frame: B46B84B171C9A5A8E1743B30369E590A207B
      Raw IMBE Frame: A5D22A00B228FF3D7083544899E064988BBC
      Raw IMBE Frame: DCEDA8FA0622BD00AC691C1FE812B503CEF6
      Raw IMBE Frame: 6C42E85DE2E8269363D981F9BE23B18AE004
      Raw IMBE Frame: 6C42A859E2E8269323D981B9BE23B18AE006
```

P25 DES-OFB Encryption



64 bit DES block

DES Block 13

DES Block 14

DES Block 15

Voice Codeword 8

LSD

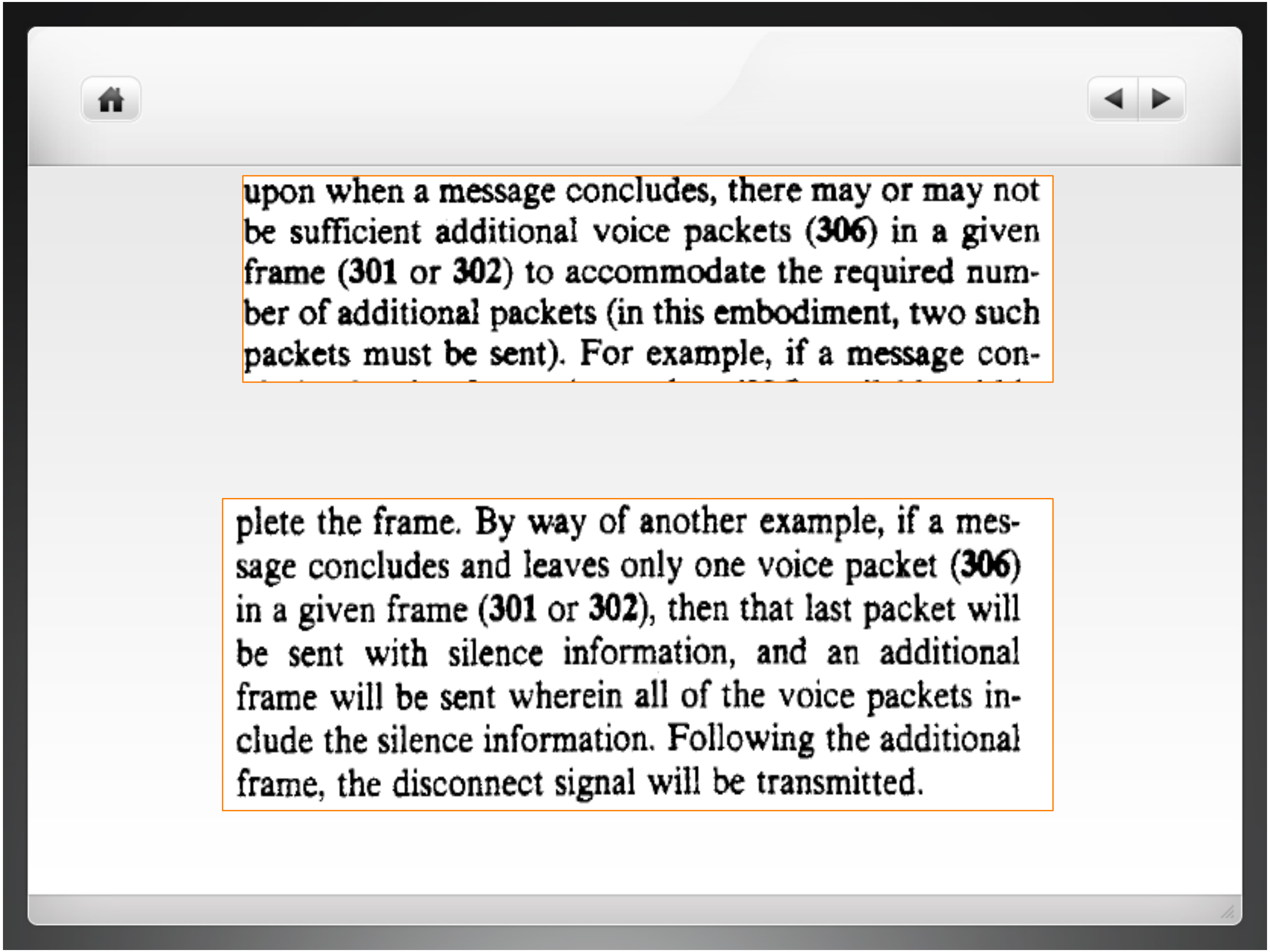
Voice Codeword 9

88 bit IMBE codeword

Unknown

But wait it gets better!

- Let say your transmission is that one that ends right on the frame boundary, or with only one silence frame at the end (at least two consecutive frames are needed)
- This doesn't give us enough Known Plaintext to be able to brute force an encryption key..
- Enter US Patent 5,220,565 – "Selective transmission of encoded voice information representing silence" – Assigned to Motorola Solutions Incorporated and was last updated in March 2011

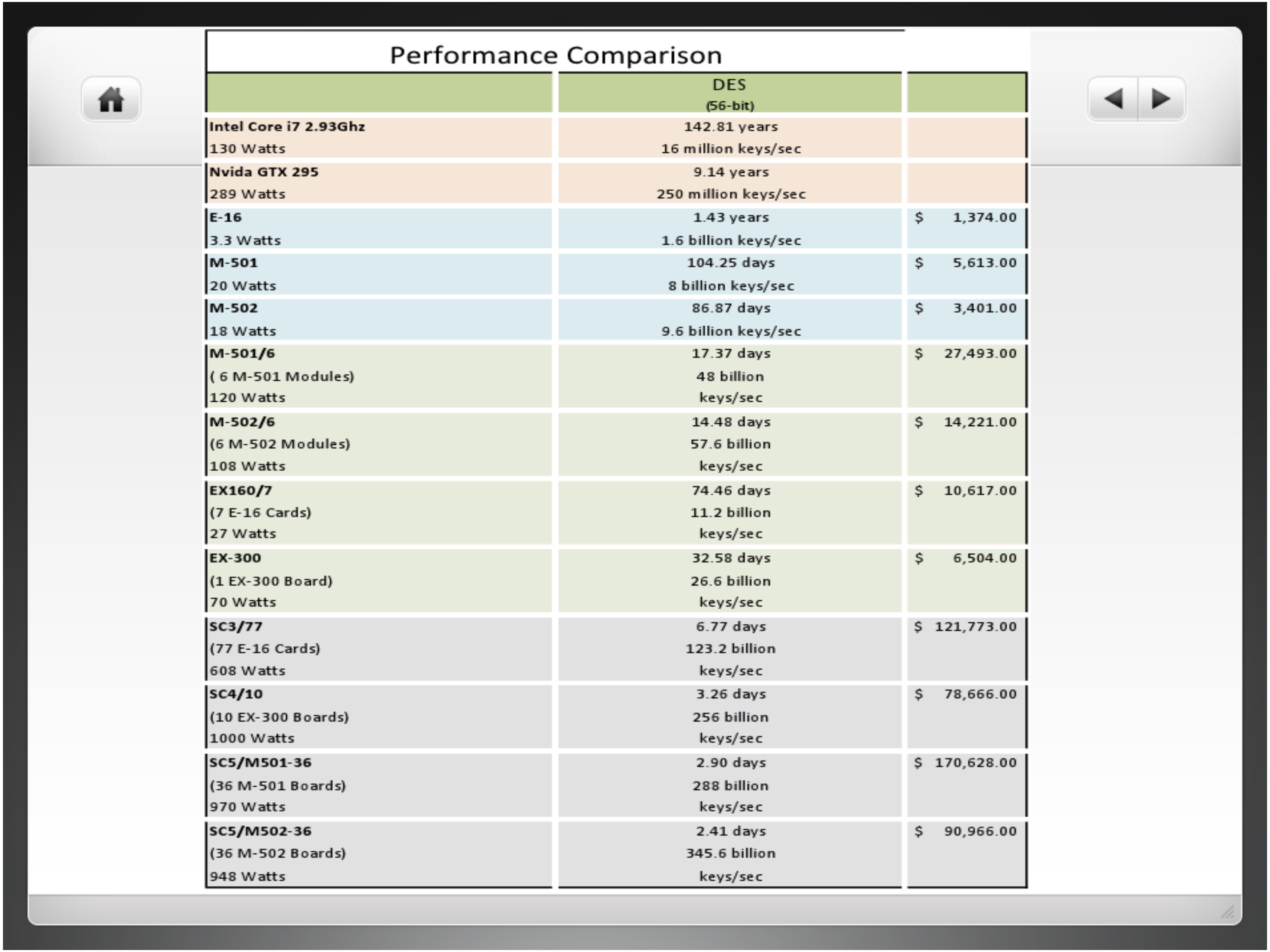


upon when a message concludes, there may or may not be sufficient additional voice packets (**306**) in a given frame (**301** or **302**) to accommodate the required number of additional packets (in this embodiment, two such packets must be sent). For example, if a message con-

plete the frame. By way of another example, if a message concludes and leaves only one voice packet (**306**) in a given frame (**301** or **302**), then that last packet will be sent with silence information, and an additional frame will be sent wherein all of the voice packets include the silence information. Following the additional frame, the disconnect signal will be transmitted.

DES Key recovery with FPGAs

- DES has been broken publicly since 1997 – there were a number of high profile projects that documented it.
  - DESCHALL
  - EFF’s Deep Crack
  - COPACOBANA
- These days, you can purchase off the shelf solutions that will brute force DES
- However – this comes at great expense. DES may be broken, but that doesn’t mean that its cheap or easy by any means…



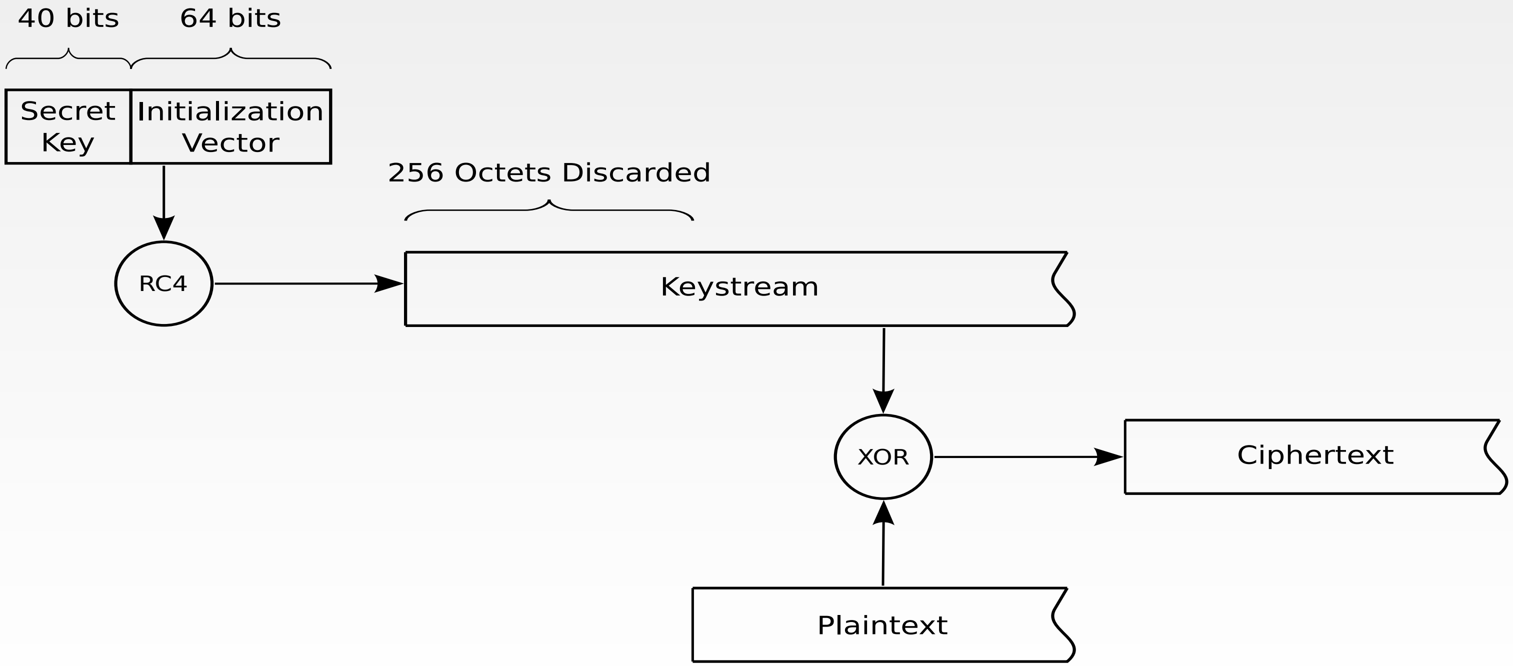
Performance Comparison

	DES (56-bit)	
Intel Core i7 2.93Ghz 130 Watts	142.81 years 16 million keys/sec	
Nvida GTX 295 289 Watts	9.14 years 250 million keys/sec	
E-16 3.3 Watts	1.43 years 1.6 billion keys/sec	$ 1,374.00
M-501 20 Watts	104.25 days 8 billion keys/sec	$ 5,613.00
M-502 18 Watts	86.87 days 9.6 billion keys/sec	$ 3,401.00
M-501/6 (6 M-501 Modules) 120 Watts	17.37 days 48 billion keys/sec	$ 27,493.00
M-502/6 (6 M-502 Modules) 108 Watts	14.48 days 57.6 billion keys/sec	$ 14,221.00
EX160/7 (7 E-16 Cards) 27 Watts	74.46 days 11.2 billion keys/sec	$ 10,617.00
EX-300 (1 EX-300 Board) 70 Watts	32.58 days 26.6 billion keys/sec	$ 6,504.00
SC3/77 (77 E-16 Cards) 608 Watts	6.77 days 123.2 billion keys/sec	$ 121,773.00
SC4/10 (10 EX-300 Boards) 1000 Watts	3.26 days 256 billion keys/sec	$ 78,666.00
SC5/M501-36 (36 M-501 Boards) 970 Watts	2.90 days 288 billion keys/sec	$ 170,628.00
SC5/M502-36 (36 M-502 Boards) 948 Watts	2.41 days 345.6 billion keys/sec	$ 90,966.00

ADP Key Recovery

- Advanced Digital Privacy (ADP):
  - Motorola proprietary cipher system
- Details of the cipher not disclosed
  - Suspected to be RC4-based
  - Key size is known – 40 bits (US export rules)
- Reverse-engineering the cipher
  - Generate known-plaintext traffic under known key
  - Capture using software-defined radio
  - Recover and match keystream at all positions to recover plaintext

ADP Cipher


40 bits
64 bits
Secret Key
Initialization Vector
256 Octets Discarded
RC4
Keystream
XOR
Ciphertext
Plaintext

RC4 Algorithm

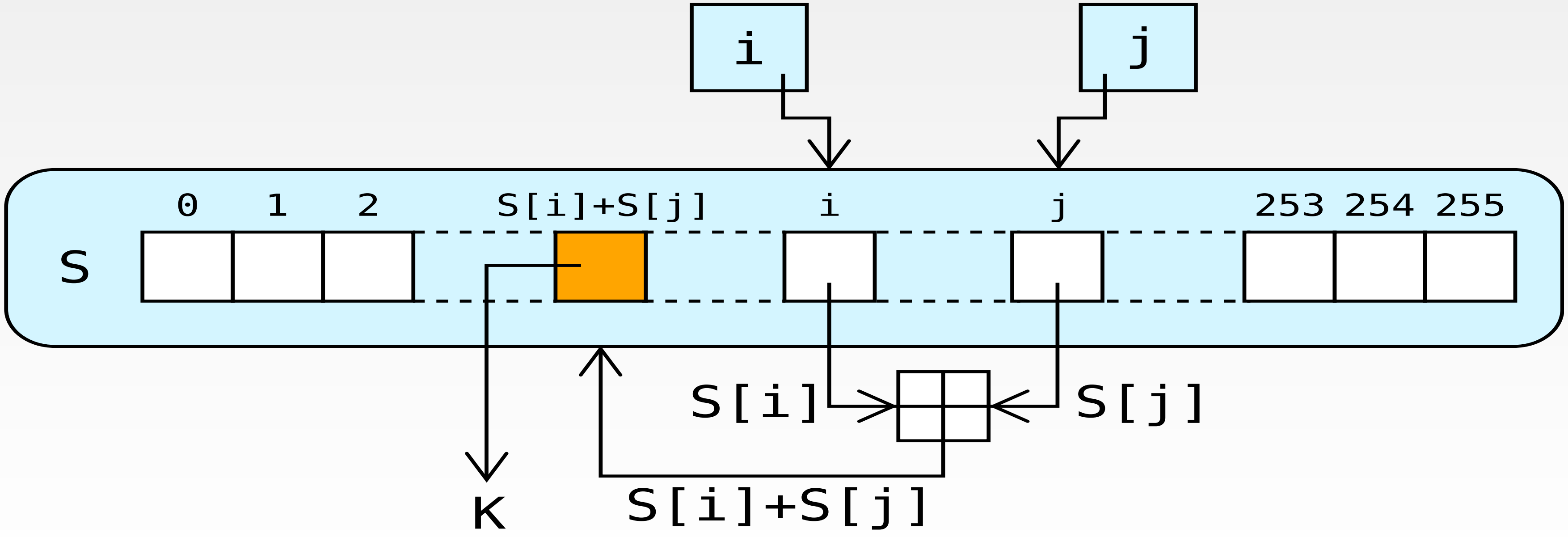
Key Scheduling Algorithm (KSA)

```
for i from 0 to 255
    s[i] := I

end

j := 0

for i from 0 to 255

    j := (j + s[i] + key[i mod keylength]) mod 256

    swap(s[i], s[j])

end
```

Pseudo Random Generation Algorithm (PRGA)

```
i := 0

j := 0

for n from 0 to KS_SZ:

    i := (i + 1) mod 256

    j := (j + s[i]) mod 256

    swap(s[i], s[j])

    ks[n] := s[(s[i] + s[j]) mod 256]

end
```

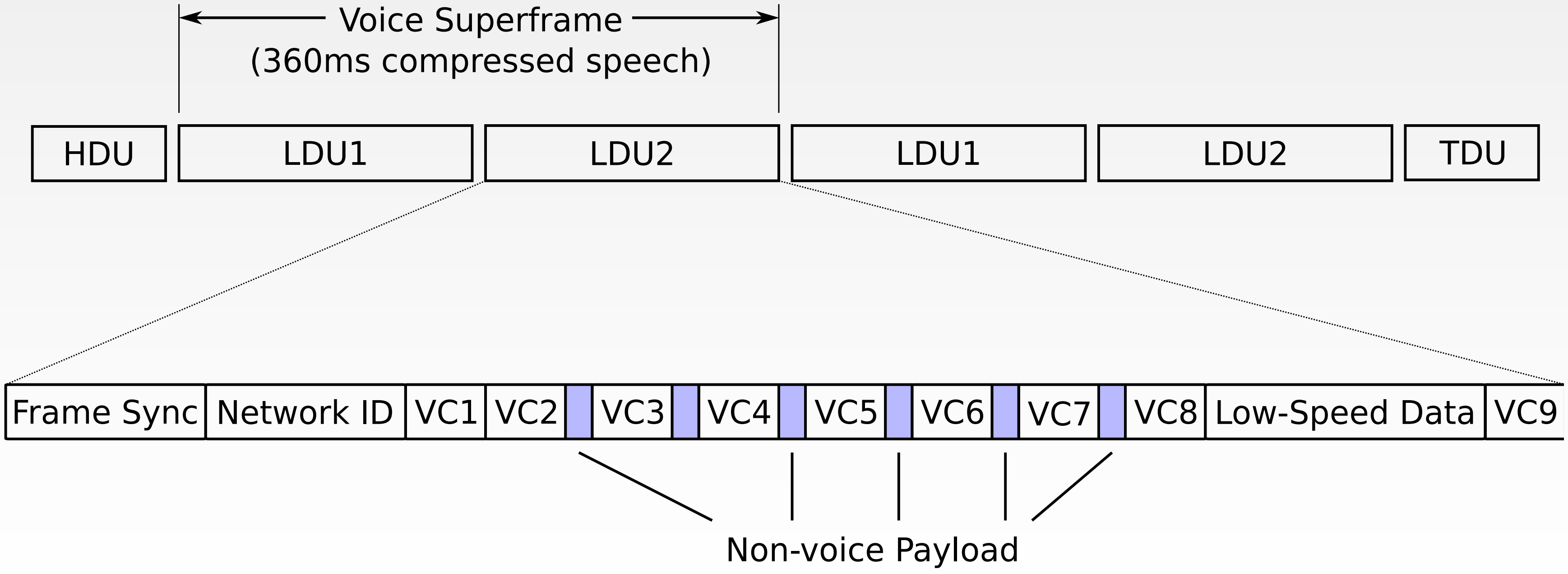
RC4 Algorithm



Weak Confidentiality – ADP Key Recovery

- IMBE vocoder has two useful properties:
  - Silence is represented using a codeword whose
- Silence occurs at easily identifiable locations:
  - Always at beginning of transmission (aka audio muting)
  - End of transmission (which silence pads transmission to next 180ms boundary)
- So we can use presence of silence to recover the keystream and brute-force the search for the key that generates that keystream

ADP Key Recovery



Voice Superframe
(360ms compressed speech)

HDU | LDU1 | LDU2 | LDU1 | LDU2 | TDU

Frame Sync | Network ID | VC1 | VC2 | VC3 | VC4 | VC5 | VC6 | VC7 | VC8 | Low-Speed Data | VC9

Non-voice Payload

Weak Confidentiality – ADP

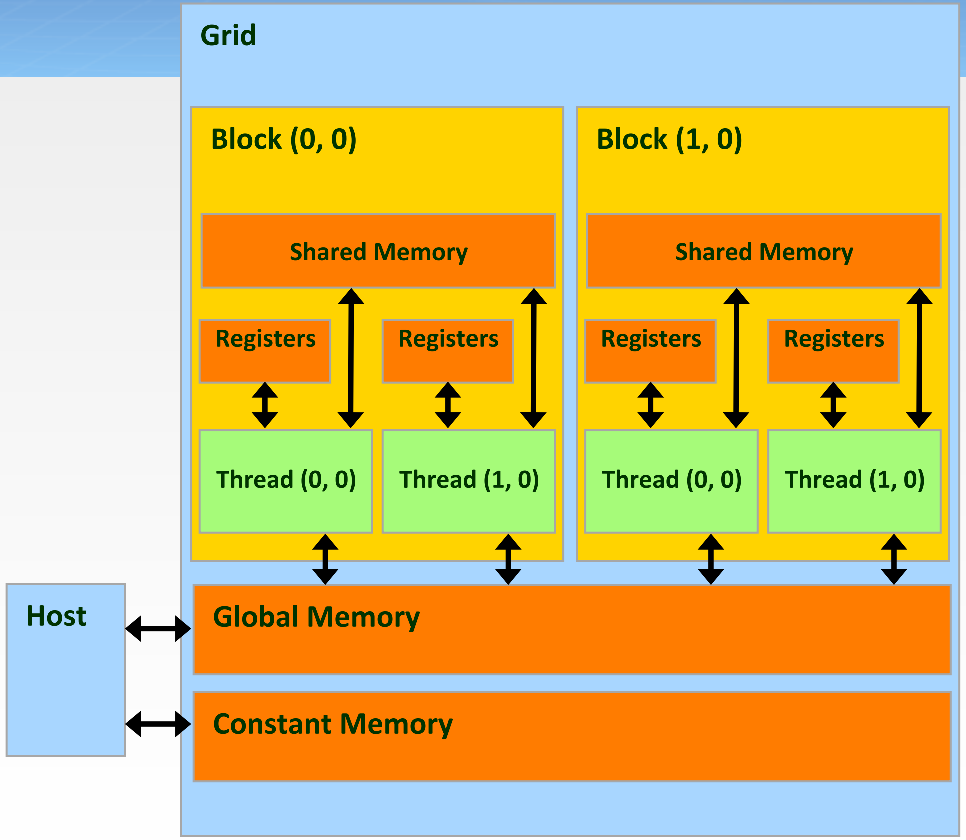
- Hardware accelerated key search –
  - FPGA about 3-4x faster than CPU
  - Large parallel FPGA search
    - 512 FPGA device
    - 40bits key space in 7 minutes
  - GPU about 3-5x faster than CPU
    - Higher cost than FPGA but many people can collaborate over Internet
    - Critical limit is not the number of GPUs but efficiency of algorithm

ADP – Key Recovery Strategies

- Hardware accelerated key search –
  - FPGA about 3-4x faster than CPU
  - Large parallel FPGA search
    - 512 FPGA device
    - 40bits key space in 7 minutes
  - GPU about 3-5x faster than CPU
    - Comparable with FPGA
    - Much more widely available
    - Can be used in distributed.net type key searches

GPU Architecture

- GPU
  - Multiple SMPs per GPU
  - Massive multi-processing
  - For most efficient use recommended 100s/1000s of active threads per SMP
- GPU Architecture
  - Single instruction/multiple data
  - Instruction-level scheduling
  - Many registers/SMP
  - 16384 registers per SMP, 16K shared memory
  - Huge performance penalties concerning memory access



RC4 + GPU

- Suitable for GPU?
  - Data parallel
  - Array indexing + integer arithmetic
  - No branch divergence
  - Positive result in the literature:
    - Performs 3x – 5x as fast as CPU
    - Similar results obtained in our initial implementation but this is well below the potential which could be 10x-15x faster

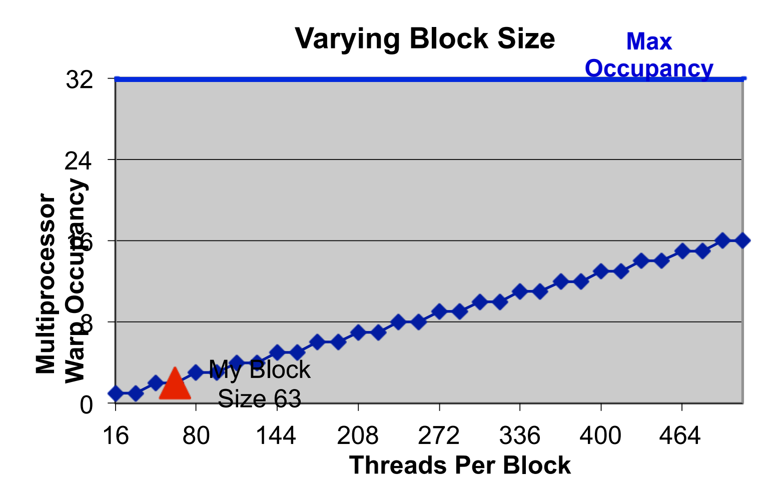
Changxin Li and Hongwei Wu and Shifeng Chen and Xiaochao Li and Donghui Guo, *Efficient implementation for MD5-RC4 encryption with GPU and CUDA*, Proceedings 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication (ASID 2009), p160-170, August 2009

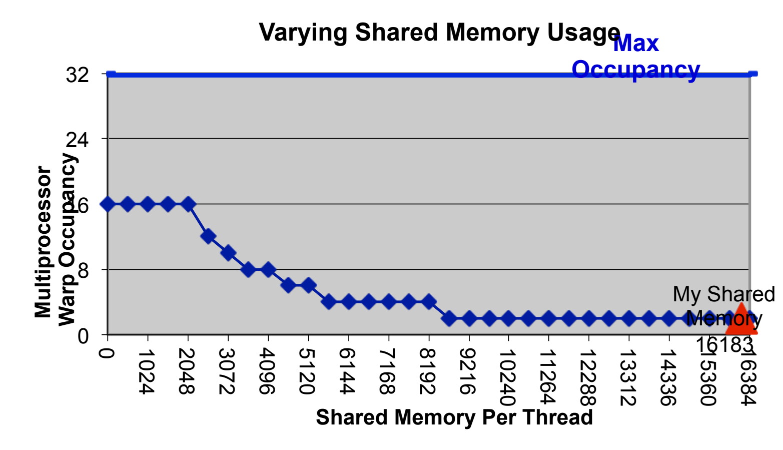
RC4 Performance

Processor	Clock Speed (GHz)	# CPUs	# Cores/CPU	VC1 Keys/ s(x 10^6)	VC9 Keys/ s(x 10^6)
Core 2 Duo	1.2	1	2	.270	.213
Core 2 Duo	2.2	1	2	.475	.378
Opteron	2.6	64	2	.375	.288
Core i7	2.6	1	2	.632	.513
Tesla	1.3	2	240	1.937	1.668

RC4 Performance

- RC4 state memory
  - 256 octets
  - Unpredictable access pattern
  - Changed for each octet of keystream
- Shared memory limit 16384 – *overhead* bytes
  - 63 threads at most!
  - 1/16 GPU occupancy = 1/16 efficient!
  - Increasing blocks available makes no difference

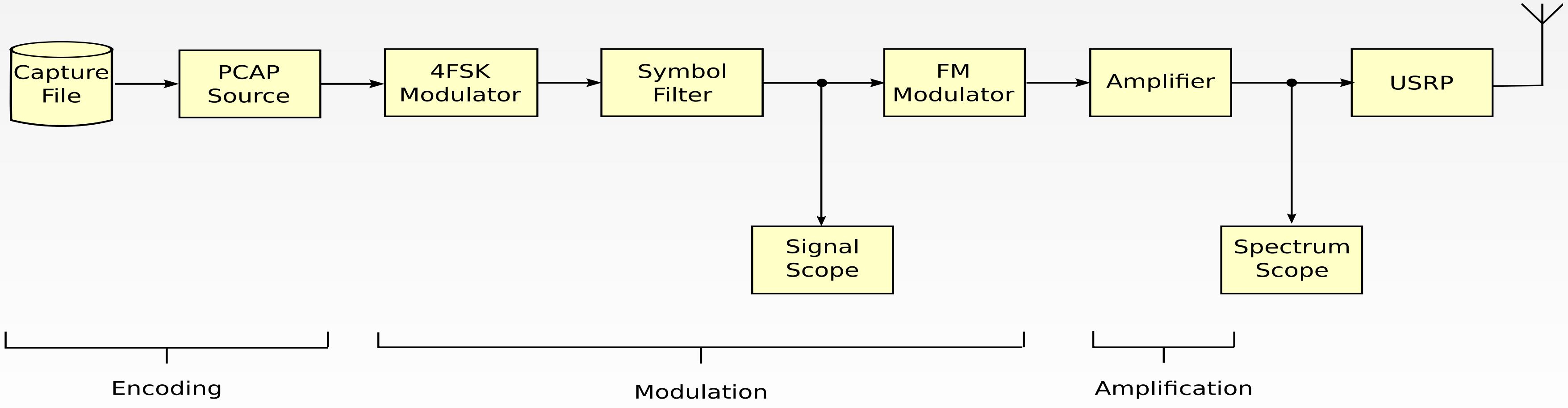




P25 Un-authenticated Inhibit Attack

- OK - so from what we know already..
  - It is mandatory to achieve conformance..
  - Radio receiving stun becomes inoperative
  - This feature does not use any authentication
  - Operates by sending trunking packet sent to a target radio
  - Easily spoofed by an adversary/attacker..
    - Since there's no authentication, we just specify a target radio ID (0xFFFFFF == ALL_CALL… potential for serious mischief?!)
    - Control Channel needs to be overpowered in trunking mode

OP25 Transmitter – block diagram



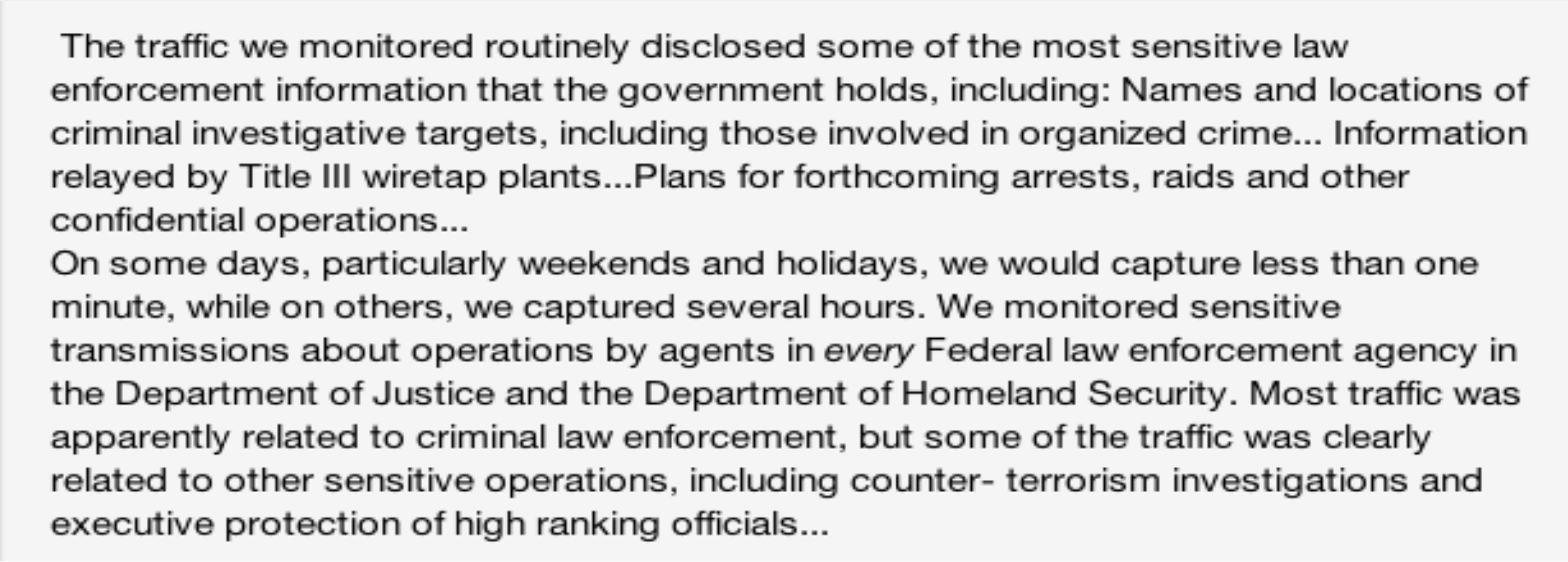
OP25 transmitter – This is what we need to transmit Inhibit packets..

- In addition to the C4FM modulator, filters, power amplifier, FM modulators we need..
- An Encoder!!
  - This creates the actual packets that we will transmit
  - This is where we can specify target Radio ID
  - We can also spoof the source ID
  - Also specify the target NID (network ID)
- Packet creator and framer
- Rate 1/2 trellis encoder

Accidental Clear transmissions on secure channel

- In a cryptonet, a bunch of encrypted operators hear unencrypted transmissions too
- So you can and will hear operators operating in the clear even though their peers are encrypted – since to the end user there is no perceived difference
- No alerting that radios are transmitting in the clear
- Therefore messages inadvertently transmitted in the clear

Accidental Clear transmissions on secure channel



The traffic we monitored routinely disclosed some of the most sensitive law enforcement information that the government holds, including: Names and locations of criminal investigative targets, including those involved in organized crime... Information relayed by Title III wiretap plants...Plans for forthcoming arrests, raids and other confidential operations...

On some days, particularly weekends and holidays, we would capture less than one minute, while on others, we captured several hours. We monitored sensitive transmissions about operations by agents in *every* Federal law enforcement agency in the Department of Justice and the Department of Homeland Security. Most traffic was apparently related to criminal law enforcement, but some of the traffic was clearly related to other sensitive operations, including counter- terrorism investigations and executive protection of high ranking officials...

"A Security Analysis of the APCO Project 25 Two-Way Radio System," Matt Blaze, Sandy Clark, Travis Goodspeed, Perry Metzger, Zachary Wasserman, and Kevin Xu.

Accidental Clear transmissions on secure channel

- Ironically the technology used to circumvent this is already there – its called strapping however no one uses it..
- Why?
- Legacy attitudes that radio will be rendered useless if the keys get "dropped"

Unencrypted Metadata attack

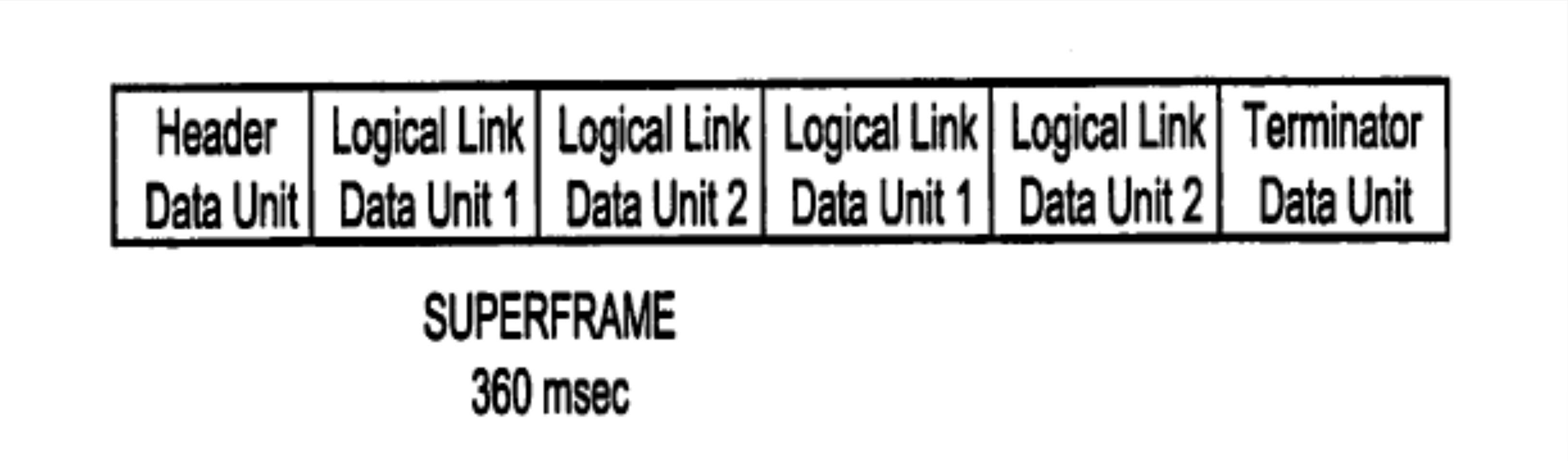
- In the HDU and LDU2 packets, ENC_SYNC information is transmitted.
- This is where the encryption algorithm in use is sent:-
  - 0x80 for clear
  - 0x81 for DES-OFB
  - 0x82 for 3DES
  - 0x84 for AES-256
  - And some proprietary algorithms too..

Unencrypted Metadata attack

- The proprietary ALGOs are:-
  - ADP (which is really just RC4)
  - DES-XL
  - DVP-XL

  - Anyway, I digress! Back on track!

Unencrypted Metadata attack



Header Data Unit	Logical Link Data Unit 1	Logical Link Data Unit 2	Logical Link Data Unit 1	Logical Link Data Unit 2	Terminator Data Unit

SUPERFRAME
360 msec

Figure 1: Data Frame structure (from Project 25 FDMA - Common Air Interface: TIA-102.BAAA-A)

Unencrypted Metadata attack

- Well why is this of any significance?
  - Unlike, say SSL, where a handshake is performance, and once trust is established session parameters are transferred..
  - P25 just blindly accepts whatever it hears..
  - While this is obviously bad from a security perspective, it leads to a very interesting and effective technical/social engineering attack..

Unencrypted Metadata attack

- A bunch of users in a group are communicating effectively – everyone is encrypted and communications are secured.
- Everyone is happy.
- Until an attacker comes along. Using the unprotected Radio ID information in the LDU1 field, (s)he can perform frequency analysis (so what) to determine who might be the network controller (for example)
- The attacker cannot hear the encrypted messages.. BUT!

Unencrypted Metadata attack

- Once he has observed the encryption algorithm, Key ID and Network ID – which are all transmitted in the clear as unprotected metadata, he can then perform a spoofing attack.

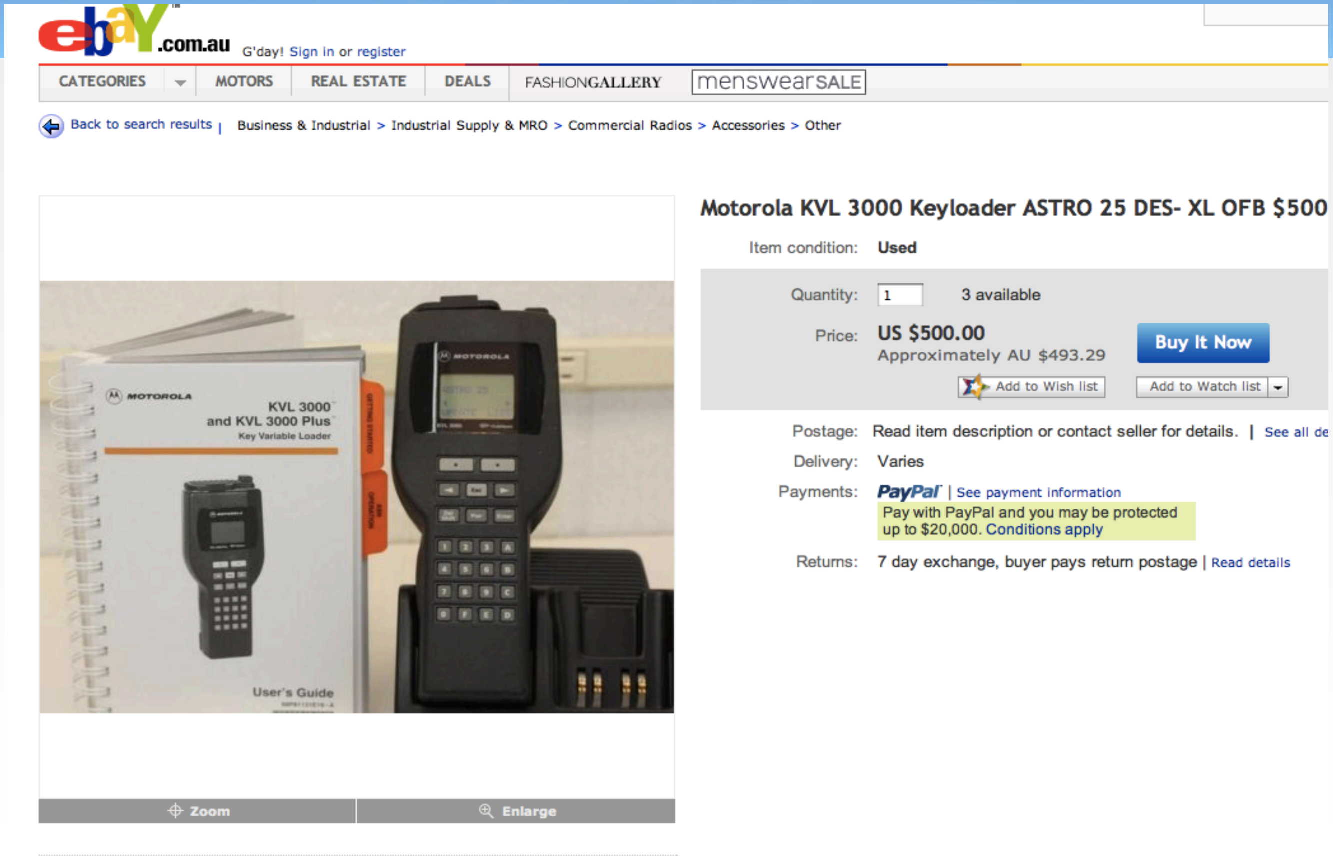
- The attacker transmits using a known existing radio ID (to emulate a valid user) he also transmits using encryption, using the SAME Algorithm ID and the SAME Key ID, but with a totally different Key Variable!

- Because all the radios blindly accept the encryption metadata as true, the radios unmute, and try to decrypt the encrypted message with their own key (which differs from the attackers)

Unencrypted Metadata attack

- As a result, the end user gets the “crickets in a blender noise”
- The users all assume there is a problem with the encryption, and then turn their encryption off!!

Keyloader Physical security



Motorola KVL 3000 Keyloader ASTRO 25 DES- XL OFB $500

Item condition: Used

Quantity: 1 3 available

Price: US $500.00
Approximately AU $493.29

Postage: Read item description or contact seller for details. | See all de

Delivery: Varies

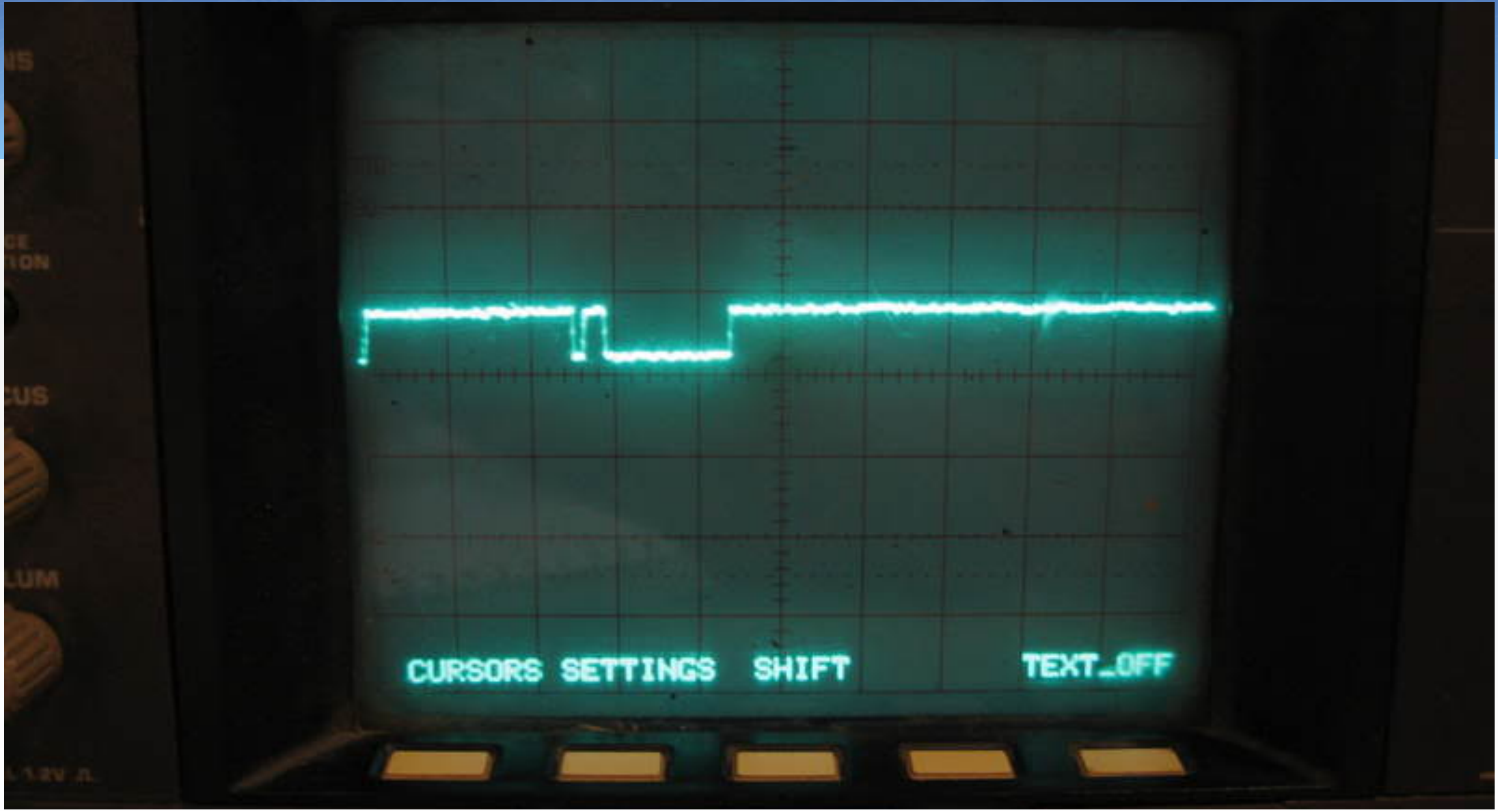
Payments: PayPal | See payment information
Pay with PayPal and you may be protected up to $20,000. Conditions apply

Returns: 7 day exchange, buyer pays return postage | Read details

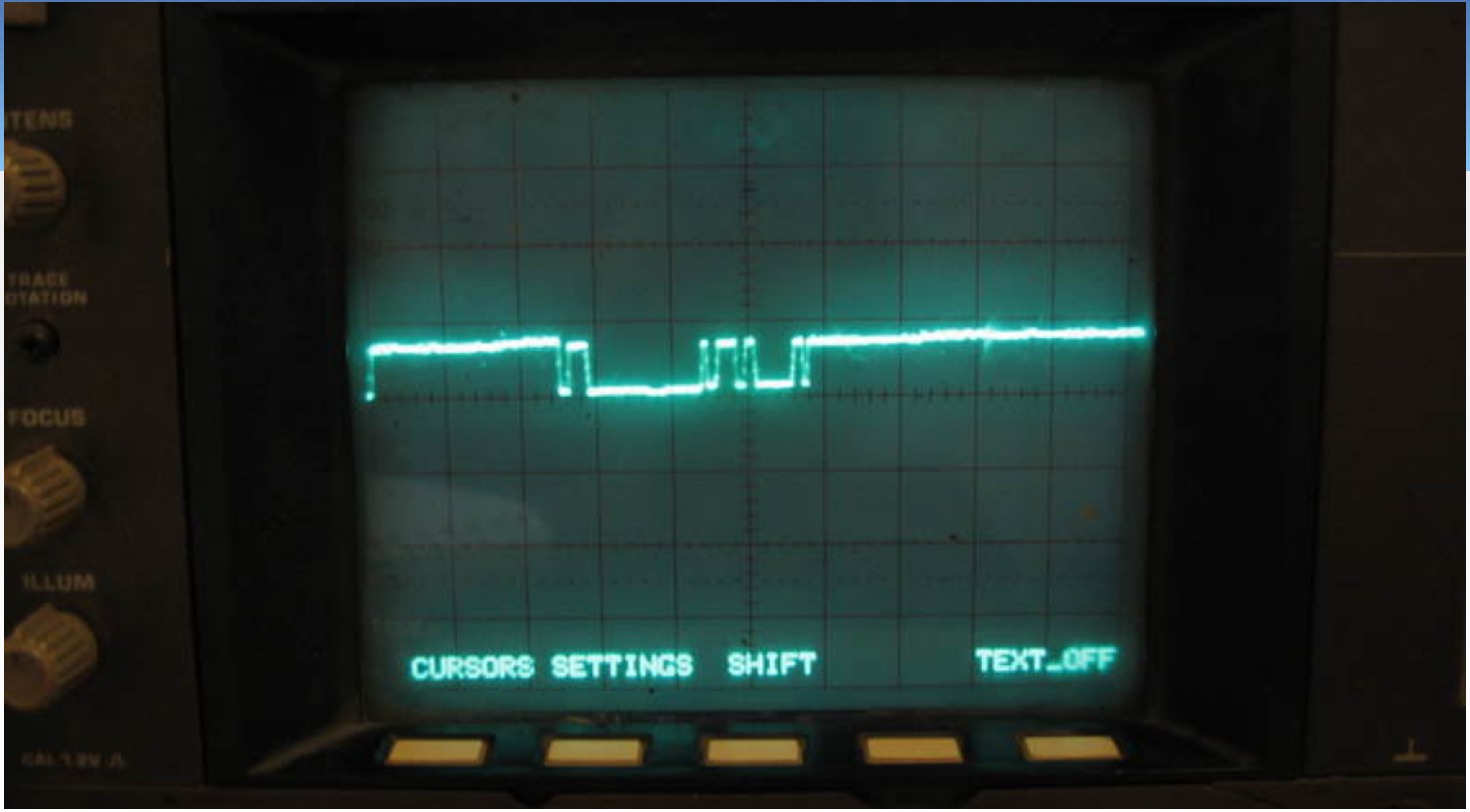
Keyloader Physical security

- Keys are transmitted in the clear on the wire
- The radio has a single bidirectional port on the side that communicates with a Keyfill device – operates at 4000 baud and has a relatively simply format.
- Encryption on the wire is supported, but only in next generation radios – seems to not be implemented at all right now?

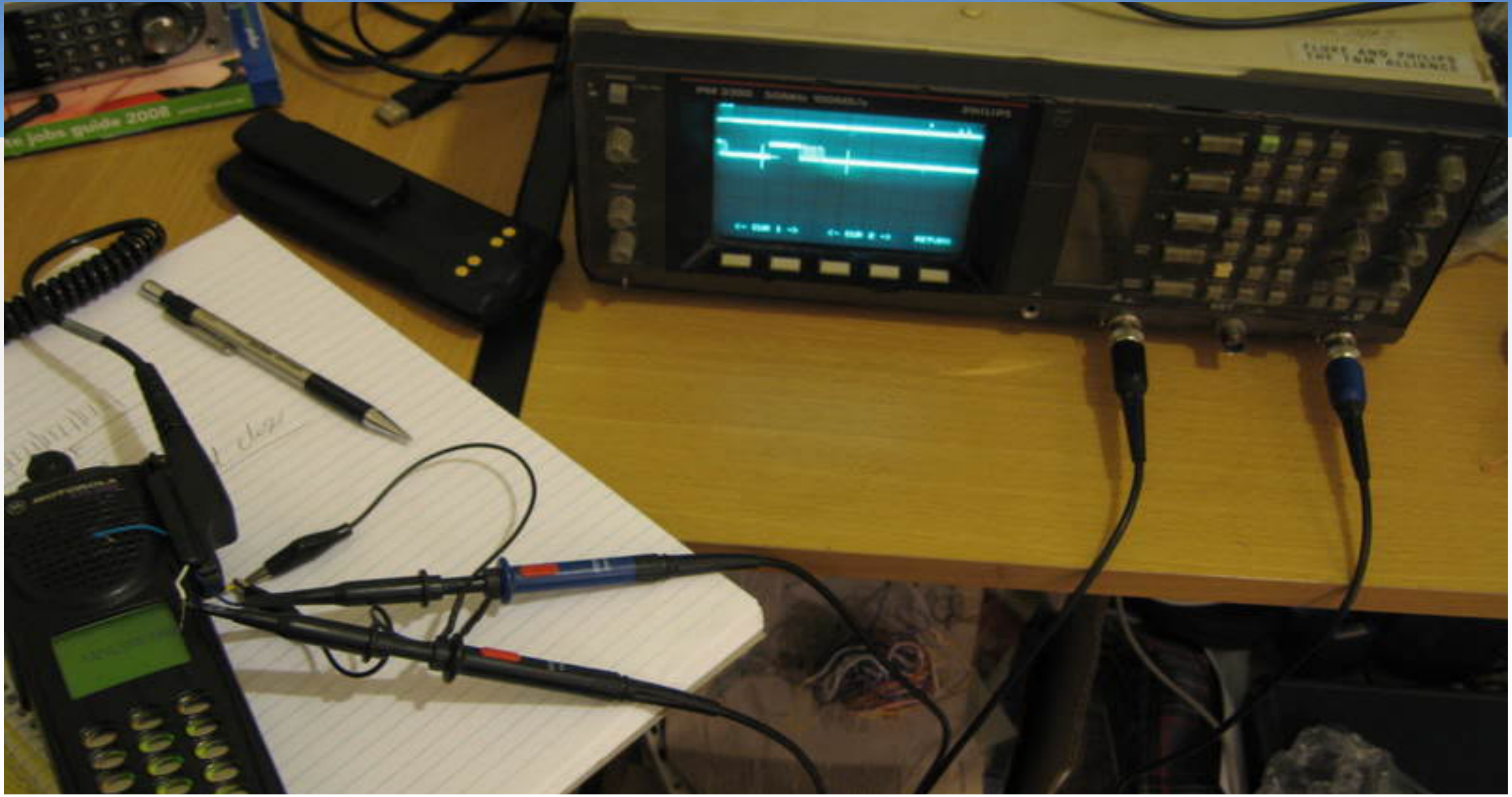
P25KFD Protocol – Simulated by ATMega8



Plug a radio in and we get a handshake!

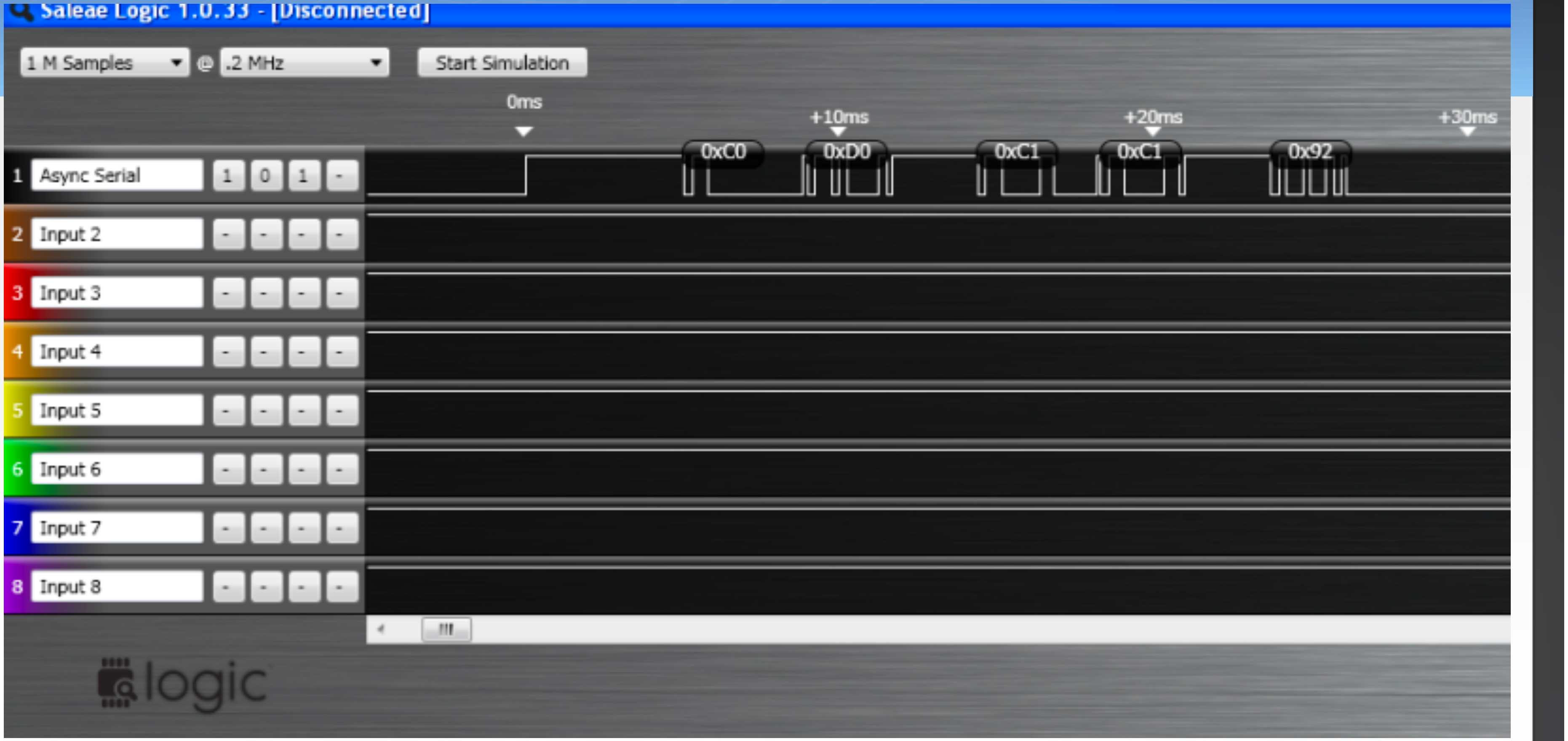


Initial investigations with DSO and wire!



More Mess!

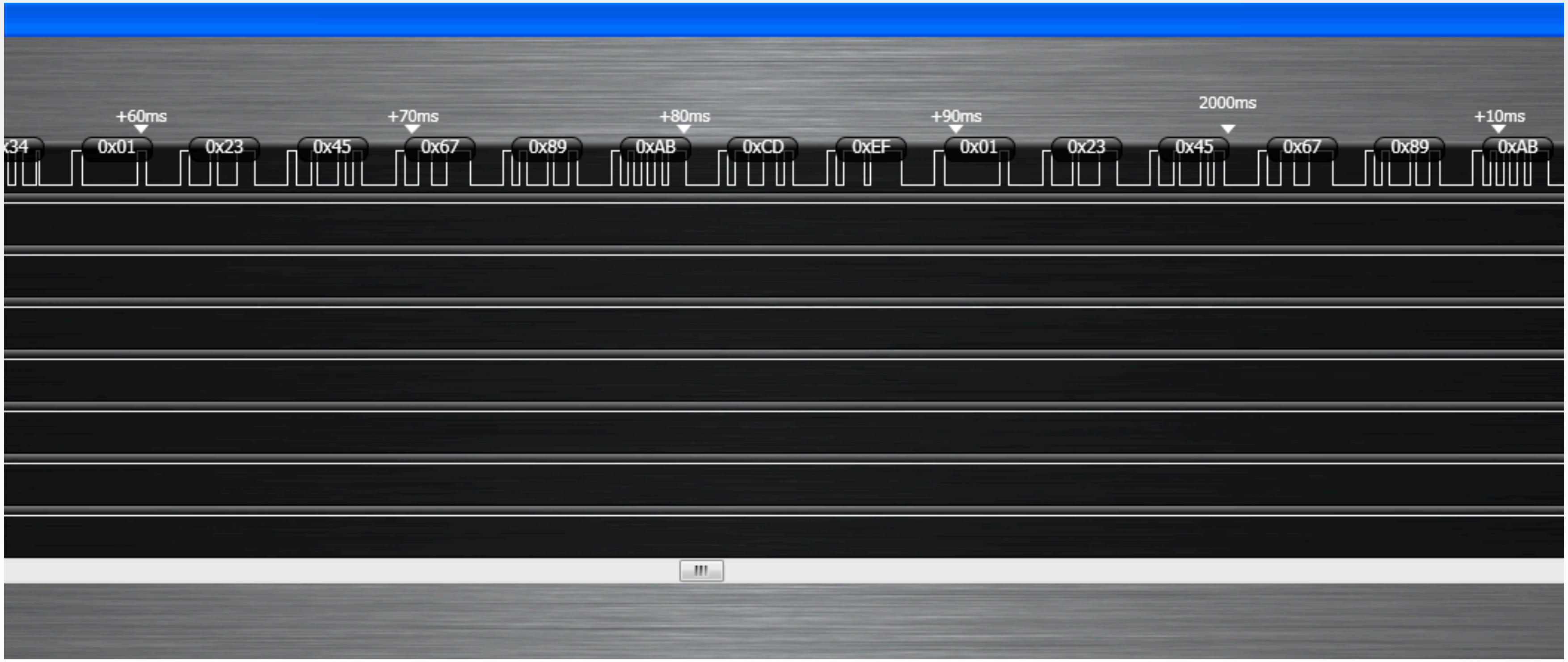
Later Attempts – Using a logic analyzer



Saleae Logic 1.0.33 - [Disconnected]

1 M Samples @ .2 MHz

Aha! There's our test key! 0x12 0x34 0x56..



OK so that seems to be promising..

- Confirms that all security is gone once the unit is physically compromised (of course..)
- Also – Keyloaders not only load keys into radios, they can load keys into other keyloaders too!
- Second hand KVL3000s with basic algorithms (DES-OFB) can be had from eBay USA for as little as $300USD!

That all works in theory, but..

- You still need physical access to the device..
- So in our opinion its not a “real” attack – Obviously its not a protocol weakness, and these devices in the real world are probably locked down in safes, probably with two people each knowing half the code…

Keyloaders

- There is no inherent security in the keyloader interface
- Authentication would be good – a cryptographic challenge/response could be used, and then keys sent over a secure channel once its been negotiated
- This could use the same authentication mechanisms as the P25 authentication scheme mentioned before – this way only radios that are authorized can be keyloaded.

Crypto Paper – Securecomm 2011

- All research on Cryptographic weaknesses and DoS (“The Inhibit Attack”) has been published in a research paper sponsored by NICTA and delivered at Securecomm 2011

NICTA

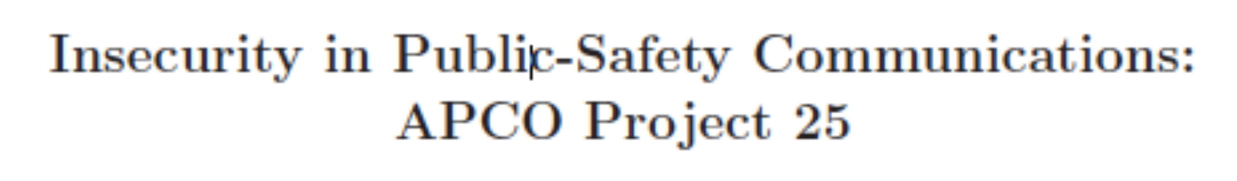
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners

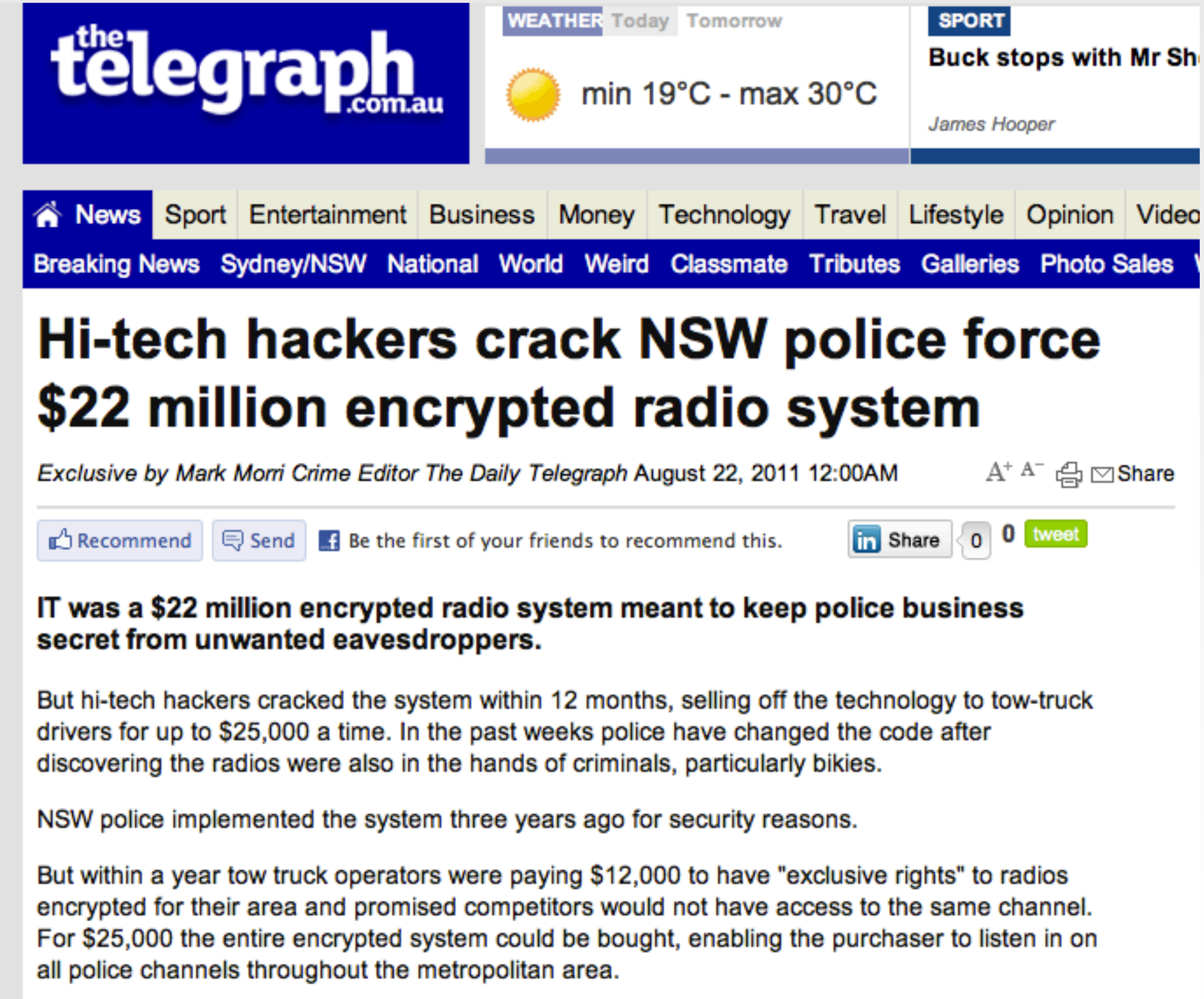
ANU THE AUSTRALIAN NATIONAL UNIVERSITY
UNSW THE UNIVERSITY OF NEW SOUTH WALES
NSW GOVERNMENT Industry & Investment
Victoria The Place To Be
THE UNIVERSITY OF MELBOURNE
THE UNIVERSITY OF SYDNEY
Queensland Government
Griffith UNIVERSITY
QUT Queensland University of Technology
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Crypto Paper – Securecomm 2011

- The paper is available on NICTA's website – www.NICTA.com.au
- Click on the publications link and enter "Matthew Robert" or "Stephen Glass" or "APCO Project 25" or the title:-



Insecurity in Public-Safety Communications:
APCO Project 25


the telegraph.com.au
WEATHER Today Tomorrow
min 19°C - max 30°C
SPORT
Buck stops with Mr Sh
James Hooper
News Sport Entertainment Business Money Technology Travel Lifestyle Opinion Video
Breaking News Sydney/NSW National World Weird Classmate Tributes Galleries Photo Sales
A+ A− Share
Recommend Send Be the first of your friends to recommend this.
Share 0 0 tweet

Hi-tech hackers crack NSW police force $22 million encrypted radio system

Exclusive by Mark Morri Crime Editor The Daily Telegraph August 22, 2011 12:00AM

IT was a $22 million encrypted radio system meant to keep police business secret from unwanted eavesdroppers.

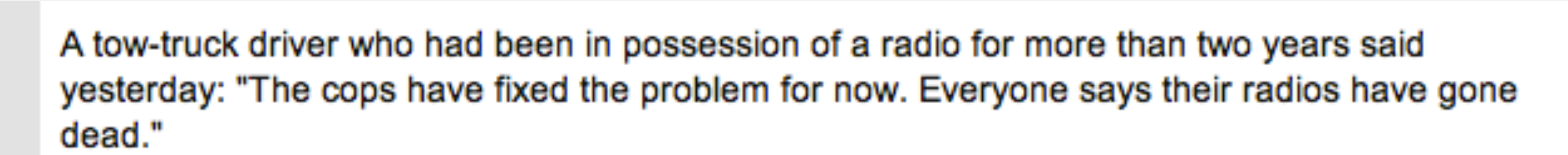
But hi-tech hackers cracked the system within 12 months, selling off the technology to tow-truck drivers for up to $25,000 a time. In the past weeks police have changed the code after discovering the radios were also in the hands of criminals, particularly bikies.

NSW police implemented the system three years ago for security reasons.

But within a year tow truck operators were paying $12,000 to have "exclusive rights" to radios encrypted for their area and promised competitors would not have access to the same channel. For $25,000 the entire encrypted system could be bought, enabling the purchaser to listen in on all police channels throughout the metropolitan area.

Yep, its disappointing to see bad things happen..

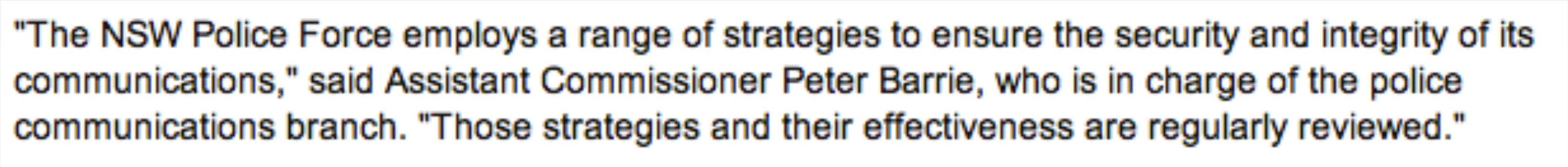
- And this particular problem pre-dates our research:-



> A tow-truck driver who had been in possession of a radio for more than two years said yesterday: "The cops have fixed the problem for now. Everyone says their radios have gone dead."

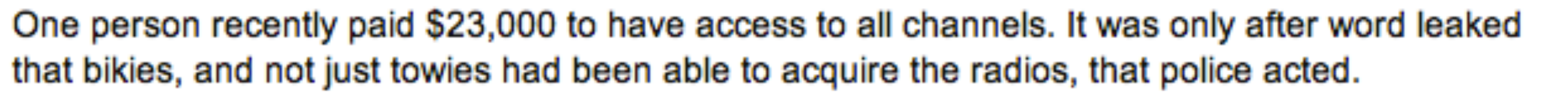
- That’s 2 years prior to that article being published in August 2011, which means they’ve already had issues since before August 2009
- Our DES-OFB research was originally made public at RUXCON 2010…

Quote one



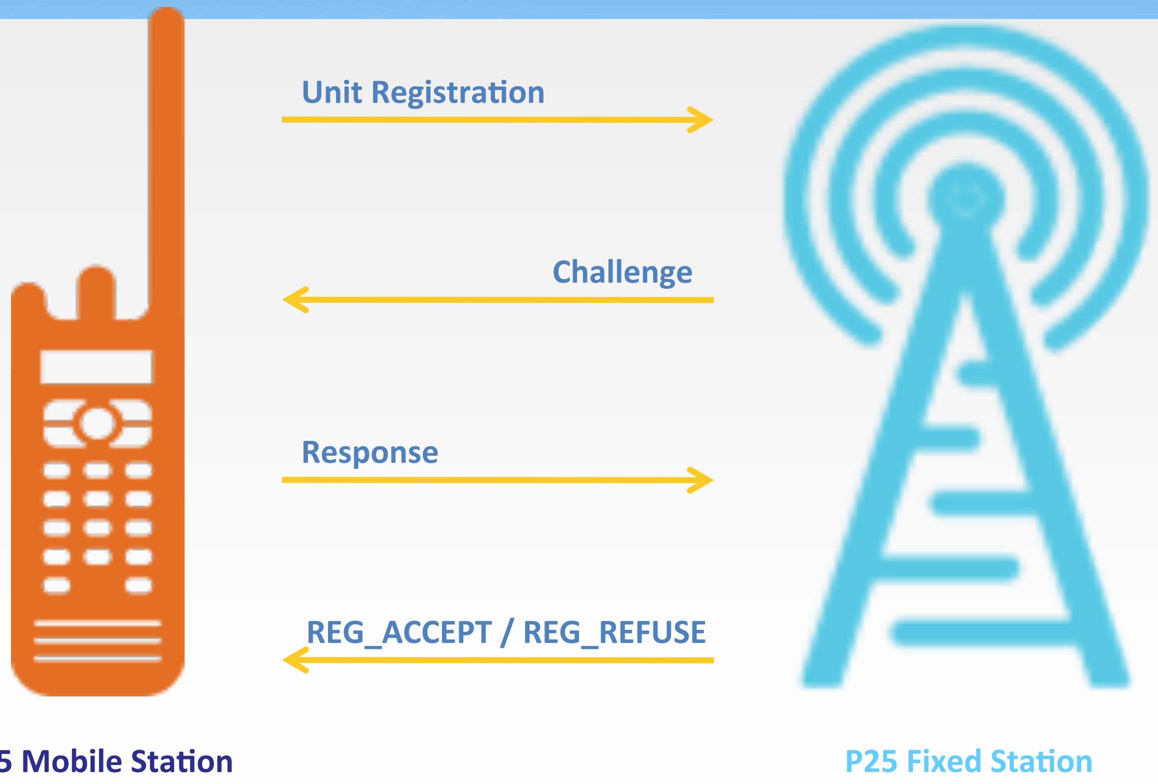
"The NSW Police Force employs a range of strategies to ensure the security and integrity of its communications," said Assistant Commissioner Peter Barrie, who is in charge of the police communications branch. "Those strategies and their effectiveness are regularly reviewed."

Quote two



One person recently paid $23,000 to have access to all channels. It was only after word leaked that bikies, and not just towies had been able to acquire the radios, that police acted.

P25 Authentication – Preventing Un-authorised radios on trunked networks



Unit Registration

Challenge

Response

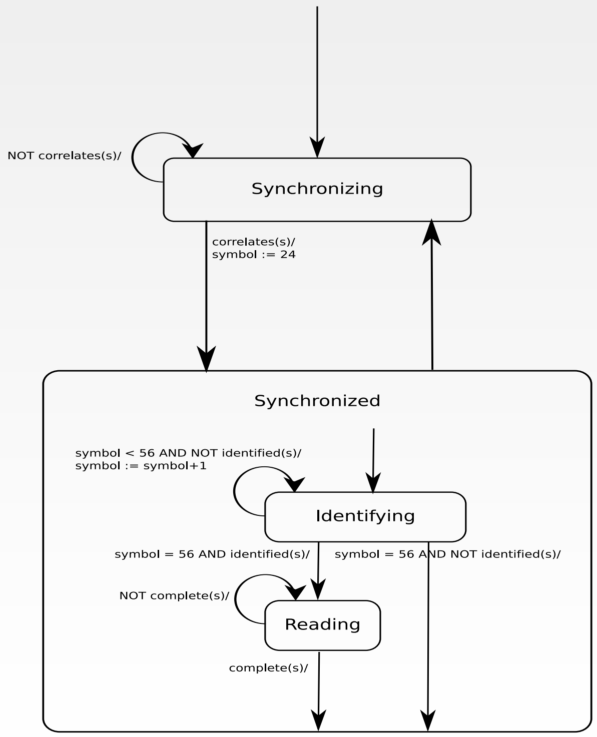
REG_ACCEPT / REG_REFUSE

P25 Mobile Station

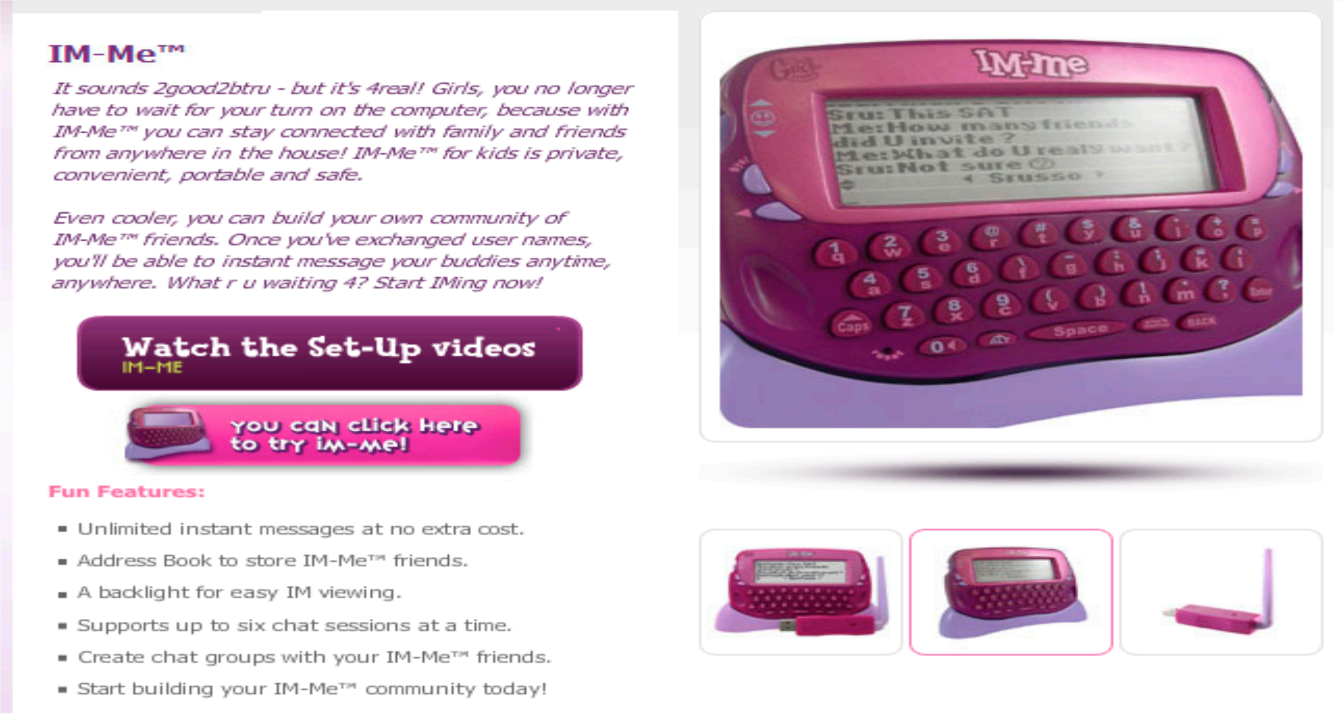
P25 Fixed Station

Sub-Frame Jamming

- The FS is used to identify *start* of frame using a simple bit correlation scheme
- The DUID is used to identify the *type* and *receiver* of the frame
- A simple state-machine is used to reconstruct the frame body based on DUID
- Damaged DUID makes entire frame unreadable
  - BCH (16, 63, 7) forward error correction



Sub-Frame Jamming - $30 worth of toys as the RF hardware!



IM-Me™

It sounds 2good2btru - but it's 4real! Girls, you no longer have to wait for your turn on the computer, because with IM-Me™ you can stay connected with family and friends from anywhere in the house! IM-Me™ for kids is private, convenient, portable and safe.

Even cooler, you can build your own community of IM-Me™ friends. Once you've exchanged user names, you'll be able to instant message your buddies anytime, anywhere. What r u waiting 4? Start IMing now!

Fun Features:

- Unlimited instant messages at no extra cost.
- Address Book to store IM-Me™ friends.
- A backlight for easy IM viewing.
- Supports up to six chat sessions at a time.
- Create chat groups with your IM-Me™ friends.
- Start building your IM-Me™ community today!

The End!

- Thanks for listening!

- Be safe, put on your licence!